콤퓨터전자도서

UNIX, LINUX
명령어사전

정보산업지도국

머 리 글

위대한 령도자 **김정일**동지께서는 다음과 같이 지적하시였다.

《현 시대는 과학과 기술의 시대, 콤퓨터시대입니다.》

경애하는 장군님께서 지적하신바와 같이 현 시대는 과학과 기술의 시대, 콤퓨터시대이다.

정보산업시대의 요구에 맞게 나라의 정보과학기술을 하루빨리 세계적수준으로 끌어올리자면 콤퓨터기술을 끊임없이 발전시켜야 한다.

최근 세계적으로 가장 주목을 끌고있는것은 Linux에 대한 연구이다.

Linux에 대한 관심이 날을 따라 높아가고 그에 대한 연구사업이 활발히 진행되여 현재 여러 종류의 Linux배포판들이 만들어져 널리 리용되고있다.

편집위원회는 정보과학기술분야에서 최첨단과학기술을 빨리 받아들이며 세계적수준의 연구성과들을 이룩해나가고있는 콤퓨터기술자, 전문가들에게 도움을 주기 위하여 전자사전《UNIX, Linux명령어사전》을 내놓게 되였다.

《UNIX, Linux 명령어사전》은 총 5개장으로 구성되여있다.

제1장에서는 UNIX와 Linux에서 많이 사용되는 명령어들을 정확히 리해하고 활용할수 있도록 실례를 들어 구체적으로 설명하였다.

제2장에서는 Linux의 기본봉사를 담당하는 데몬에 대하여 서술하였다.

제3장에서는 Linux환경을 보다 편리하게 해주는 Shell스크립트에 대하여 설명하였다.

제4장에서는 vi편집기에 대한 설명을 주었다.

제5장에서는 Linux를 최고의 개발환경으로 만들어주는 번역기에 대하여 설명하였다.

편집위원회는 《UNIX, Linux명령어사전》이 새로 편찬되는것만큼 부족점이 있으리라고 생각하면서 독자여러분들의 좋은 의견을 바란다.

차　　례

제 1 장　일반명령문 (COMMANDS)

A

alias(명령문을 간단히 사용하기) 2
apm(학습장형콤퓨터의 전원관리) 5
appletviwer(Java 애플리트의 실행) 7
apropos(관련명령의 찾기) 9
arch(CPU 형의 보기).................................... 11
arp(체계의 MAC 주소확인).................................... 13
arping(망련결확인) 16
at(작업예약).................................... 18
atq(예약작업보기) 23
atrm(예약작업삭제) 24
awk(패턴처리언어) 26
a2p(awk-perl 변환) 33

B

badblocks(불량블로크의 검사) 35
banner(문자확대출력) 38
basename(파일이름추출).................................... 40
batch(명령실행의 예약).................................... 42
bc(본문계산기) 44
biff(우편도착의 알림).................................... 46
bzip2(압축명령문) 47

C

cal(력서보기) 50
cardctl(PCMCIA 기판의 관리).................................... 53
cardmgr(PCMCIA 장치관리자).................................... 56

cat(파일보기)..58
cd(파일등록부이동)....................................62
cfdisk(구획설정)......................................63
chage(암호유효기일의 지정)............................65
chattr(속성변경)......................................68
chfn(사용자정보의 변경)...............................70
chgrp(집단변경).......................................72
chkconfig(구동프로그람설정)...........................74
chmod(권한변경).......................................77
chown(소유자변경).....................................84
chroot(가상뿌리등록부)................................87
chsh(쉘변경)..88
clear(화면지우기).....................................91
cmp(파일비교)...92
col(개행문자변환).....................................93
colcrt(밑줄(__)문자변환)..............................94
colrm(특정렬의 삭제)..................................96
column(파일정렬)......................................98
comm(한행씩 비교).....................................99
compress(파일압축)...................................101
cp(파일복사)...103
cpio(사본만들기).....................................105
crontab(crontab 관리)................................108
csplit(파일분할).....................................111
cut(항목의 골라보기).................................114

D

date(날자보기)......................................116
dd(블로크단위의 복사)................................119
debugfs(파일체계조종)................................122
depmod(모듈의존성의 확인)............................126

df(구획의 남은 용량을 보기) 127
diff(파일의 다른 점을 비교) 129
dirname(경로이름보기) 133
dmesg(기동리력의 보기) 134
dnsdomainname(DNS 령역확인) 137
domainname(NIS 령역의 보기) 138
dosfsck(MS-DOS 파일체계의 복구) 140
du(디스크사용정형을 보기) 141
dumpe2fs(파일체계정보의 보기) 143
dumpkeys(건반코드값보기) 145

E

e2fsck(Linux 파일체계검사) 147
echo(환경변수출력) 150
eject(CD-ROM 자동반출) 153
env(환경변수보기) 155
ex(행편집기) 157
expand(문서변환) 158
expr(표현식평가) 160

F

false(항상 실패) 161
fdformat(플로피디스크의 초기화) 163
fdisk(구획설정) 165
fetchmail(우편을 모아오기) 171
file(파일류형확인) 173
find(파일찾기) 175
finger(사용자정보보기) 178
free(체계주기억의 보기) 180
fsck(파일체계검사) 182
ftp(ftp 봉사기접속의뢰기프로그람) 187

ftpcount(ftp 사용자확인) 192
ftpwho(ftp 접속자보기).. 193
fuser(프로쎄스식별) 194

G

gated(게트웨이경로지정) 195
gdc(gated 관리).. 197
getkeycodes(코드대응표보기)................................... 200
grep(문자렬검색)... 201
groupadd(집단추가)... 203
groupdel(집단삭제) 204
groupmod(집단정보수정) 205
groups(자기의 집단보기) 206
grpck(파일찾기)... 207
grpconv(그림자집단생성).. 208
gunzip(압축해제) 209
gzexe(실행파일압축)... 211
gzip(파일압축) 212

H

halt(체계완료) 214
hdparm(하드디스크정보의 보기 및 설정)........................ 216
head(파일앞부분의 보기)....................................... 219
host(호스트정보보기) 222
hostid(호스트 ID 지정) 224
hostname(체계이름보기).. 225

I

id(식별이름 ID 의 확인)....................................... 227
ifconfig(망대면부설정)... 229
info(하이퍼본문형의 문서페지) 234

init(초기화프로쎄스조종) 236
insmod(핵심부모듈을 불러들이기) 239
install(파일설치) 242
ipchains(Kernel 2.2.x 판본의 방화벽관리도구) 243
ipfwadm(Kernel 2.0.x 판본의 방화벽관리도구) 249
iptables(Kernel 2.4.x 판본의 방화벽관리도구) 251
ispell(철자법검사) 256

J

join(항목단위로 파일합치기) 258

K

kbd_mode(건반설정) 260
kbdrate(건반입력속도의 설정) 261
kill(프로쎄스완료) 262
killall(프로쎄스완료) 265
ksyms(핵심부 symbol의 보기) 268

L

last(체계접속시간의 보기) 269
lastlog(접속리력의 보기) 272
ldconfig(동적련결의 설정) 275
ldd(서고의존성을 보기) 277
less(파일보기) 278
lilo(Linux 기동적재기) 281
ln(파일련결명령문) 286
locale(지역정보보기) 289
locate(파일위치찾기) 291
lockfile(신호기파일생성기) 292
logger(체계리력의 기록) 294
login(체계접속명령문) 296

logname(현재 사용자이름의 보기) 298
logrotate(리력파일관리) 300
look(특정문자렬로 시작하는 행을 보기) 302
lpc(인쇄기관리자) 303
lpq(인쇄대기렬의 보기) 305
lpr(인쇄명령) 307
lprm(인쇄취소) 309
ls(파일목록보기) 311
lsattr(파일속성보기) 316
lsdev(하드웨어정보의 보기) 317
lsmod(적재된 모듈의 보기) 318
lsof(사용중인 파일들의 목록보기) 319

M

mail(우편의 송수신) 321
mailq(우편작업보기) 326
man(문서보기) 327
mattrib(MS-DOS 파일의 속성변경) 330
mbadblocks(MS-DOS 파일체계의 불량블로크 표시) 332
mcd(MS-DOS 등록부변경) 333
mcopy(Linux-DOS 간의 파일복사) 335
mdel(MS-DOS 파일삭제) 337
mdeltree(MS-DOS 파일삭제) 339
mdir(MS-DOS 파일목록보기) 341
mformat(MS-DOS 파일체계만들기) 343
minfo(MS-DOS 파일체계에서 매개변수보기) 345
mkdir(등록부만들기) 346
mke2fs(ext2 파일체계를 생성하는 명령문) 347
mkfs(Linux 파일체계생성명령문) 350
mknod(특수파일만들기) 352
mkswap(swap 령역지정) 354

mlabel(MS-DOS 디스크식별표식만들기) 355
mmd(MS-DOS 파일체계등록부의 만들기).......... 357
mmount(MS-DOS 파일체계적재) 358
mmove(MS-DOS 파일이동/이름변경) 359
modprobe(핵심부모듈적재).......... 360
more(화면단위로 파일보기).......... 364
mount(장치련결) 366
mrd(MS-DOS 등록부삭제) 370
mren(MS-DOS 파일이름변경) 371
mtype(MS-DOS 파일내용보기) 373
mv(파일이동) 374

N

namei(련결원본파일의 경로보기).......... 375
netstat(망상태보기).......... 377
newgrp(집단변경).......... 380
newusers(사용자의 추가).......... 382
nice(우선권순위변경) 383
nl(행번호매기기) 386
nohup(체계탈퇴중에도 프로그람실행).......... 388
nslookup(이름봉사기문의).......... 389

O

od(8 진수로 파일보기) 393

P

passwd(암호의 변경).......... 395
paste(파일합치기) 397
pathchk(파일이름검사) 399
pidof(PID 확인).......... 401
ping(망련결확인).......... 402

pr(파일인쇄설정) 405
praliases(우편 alias 보기) 408
printenv(환경변수보기) 409
ps(프로쎄스상태보기) 410
pstree(프로쎄스상관도) 419
pwck(암호파일검사) 421
pwconv(그림자암호의 사용) 422
pwd(현재경로보기) 423

Q

quota(디스크사용용량의 제한)................................ 424

R

ramsize(RAM 디스크크기의 확인 및 설정) 427
ranlib(서고변환) 429
rcp(원격지파일복사) 430
rdate(시간맞추기) 432
rdev(장치조사)... 433
readonly(읽기전용) 436
reboot(체계재기동)... 437
renice(우선권순위변경) 438
reset(말단을 초기화하는 명령문)............................. 439
restore(dump 예비보관파일을 복구하는 명령문) 440
rev(파일내용을 거꾸로 바꾸기) 441
rexec(원격실행).. 442
rlogin(원격호스트접속) 444
rm(파일삭제)... 446
rmail(원격우편보기) 448
rmdir(등록부삭제) 449
rmmod(모듈제거).. 450
rootflags(뿌리장치설정명령문) 452

route(경로표보기) 453
rpm(RedHat 패키쥐관리자) 456
rpm2cpio(RAM 변환명령) 467
rsh(원격호스트쉘을 사용) 468
rstat(체계식별이름정보의 제공)........................... 470
runlevel(실행준위의 보기) 471
rwall(통보문보내기)..................................... 472
rwho(접속자확인) 473

S

script(말단의 작업기록)................................. 474
scp(파일복사) 476
setfdprm(플로피디스크설정)............................... 477
setup(체계설정).. 478
sh(기본쉘의 호출) 479
shar(쉘압축편의프로그람) 480
showmount(마운트정보보기) 482
shutdown(체계끝내기) 483
slattach(직렬회선련결) 485
sleep(대기) 486
slocate(파일찾기) 487
sort(내용정렬) 489
source(환경설정의 반영)................................. 492
split(파일분할)... 493
ssh(SSH 의뢰기) 496
stat(상태보기)... 497
strace(프로그람에서 실행되는 프로쎄스의 추적) 499
strings(문자렬찾기) 500
strip(목적파일의 크기축소)................................ 502
stty(말단환경설정) 504
su(다른 식별이름의 환경을 사용하기) 509

sum(파일확인) 511
swapoff(swap 중지)... 512
swapon(swap 설정) 513
sync(주기억기의 내용을 보관)................................ 514

T

tac(파일내용을 역순으로 출력).................................. 515
tail(파일의 뒤부분보기) 517
talk(1 대 1 의 대화) 520
tar(파일묶기)... 522
tee(같은 내용을 두 경로로 동시에 출력)....................... 526
telinit(프로쎄스조종) 527
telnet(원격접속) 528
test(검사).. 531
tftp(tftp 봉사기접속의뢰기프로그람)............................ 534
top(프로쎄스상태 보여주기) 536
touch(빈 파일을 만들기)....................................... 540
tr(문자의 변환 및 삭제)...................................... 542
traceroute(파케트이동경로 보기) 544
true(참(0)값을 반환) 545
tty(말단이름보기) 546
tune2fs(파일체계설정)... 547

U

ul(밑줄긋기) 549
umount(장치련결의 해제)...................................... 550
uname(체계정보의 보기)....................................... 552
uncompress(압축해제) 556
unexpand(공백을 tab 로 변환)................................. 557
uniq(중복행의 추출) 558
uptime(평균부하의 보기)....................................... 560

useradd(사용자의 추가) 561
userdel(사용자의 제거)..................................... 563
usermod(사용자설정환경의 변경).............................. 564
users(사용자보기) 567

W

w(현재사용자의 보기)....................................... 568
wall(모든 사용자에게 통보문 보내기)......................... 569
wc(단어수의 계산) 570
whatis(명령검색)... 572
whereis(명령문경로의 검색) 573
which(파일경로의 검색) 575
who(사용자보기).. 576
whoami(유효사용자의 보기)................................... 578
write(다른 사용자에게 통보문 보내기) 579

X

xargs(표준입력을 명령으로 변환)............................. 580

Y

ypbind(NIS 구동) 582
ypcat(NIS 련결의 확인)...................................... 585
ypchfn(사용자정보의 변경) 587
ypchsh(쉘의 변경)... 588
ypmatch(key 값의 보기) 589
yppasswd(암호의 변경) 591
yppoll(NIS 정보의 보기) 592
yppush(NIS 전체의 반영) 593
ypserv(NIS 봉사기) 594

Z

zcat(압축된 파일의 내용을 보기) 596
zcmp, zdiff(압축된 파일의 내용비교)........................ 598
zdump(매 시간대에 의한 시간보기)........................ 599
zmore(압축된 본문파일의 내용보기) 600
znew(.gz 로 변환)........................ 602
ztelnet(원격접속) 603

제 2 장 데몬 (DAEMON)

atd(예약작업의 실행) 611
crond(예약작업의 실행)........................ 613
dhcpd(동적 IP 주소의 제공) 615
httpd(가장 널리 사용되는 Apache 웨브봉사기)........................ 617
lpd(행인쇄기데몬) 619
named(이름봉사기데몬)........................ 622
network(망구동스크립트)........................ 629
nfs(NFS 봉사기데몬) 631
proftpd(ftp 데몬) 632
sendmail(우편봉사기데몬)........................ 636
smb(Samba 봉사데몬)........................ 640
sshd(OpenSSH 봉사기데몬)........................ 642
syslog(체계에 발생하는 다양한 리력들을 기록하는 봉사)........................ 644
xinetd(인터네트고급데몬) 647

제 3 장 쉘프로그람작성 (SHELL PROGRAMMING)

#(주해) 657
#!(쉘선언) 658
break(정지)........................ 659
case(다중분기문) 661

dirs(기억등록부의 보기) 663
exit(완료) 664
export(변수지정) 665
for(for 순환) 666
help(내부명령문도움말) 668
history(이전 명령문) 669
if(if 조건문) 671
let(산수연산) 673
while(while 순환문) 675

제 4 장 VI편집기 (VI EDITOR)

VI 편집기의 사용해설과 명령들 .... 677~701

제 5 장 번역기 (COMPILER)

gcc 703
gdb 707
make 708
patch 711
python 713
java 714
Tcl 715
Tk 716
Qt 717
GTK + ... 718
Lex & Yacc 719

색인 720

제 1 장
일반명령문(COMMANDS)

사용자들이 UNIX, Linux체계를 얼마나 잘 사용하고있는가를 알아보기 위해 가장 좋은 방법은 그가 어떤 상황에서 어떤 명령문으로 어떻게 효과적으로 해결하는가를 확인해보는것이다. 많은 명령문을 아는것도 중요하지만 하나의 명령문이라도 주어진 상태에서 가장 효과적으로 사용하는것이 더 중요하다. 여기에서는 명령문의 사용법에 대하여 구체적으로 설명하였다.

A

alias 내부명령문

명령문을 간단히 사용하기

복잡한 명령문과 설정값을 간단히 입력할수 있는 문자렬로 치환한다.

형식 : alias name[=value]

설정값을 포함한 긴 명령문을 사용할 때마다 매번 입력함이 없이 간단한 문자렬로 치환하여 사용하기 위해 이 명령문이 리용된다. 이러한 의미에서 alias는 《별명》이라고도 불리운다.

말단으로부터 rm명령문을 리용하여 파일삭제를 할 때에 파일을 삭제하겠는가를 재차 문의하는 설정값을 사용하지 않아도 기본rm명령문이 이 설정값을 기정적으로 사용하고있는것, ls명령문을 리용하여 파일목록을 보았을 때에 색으로 구분되는것 등은 대부분의 환경에서 미리 alias로 해당한 설정값을 예약해주었기때문이다.

쉘은 내부명령인 alias와 unalias를 리용하여 이러한 예약내용을 목록에 설정하거나 해제하는것을 수행한다. 어떤 명령문이 입력되면 이 명령문의 앞에서부터 문자렬과 일치하는 alias가 있는가를 목록에서 확인하고 일치하면 원래의 명령문으로 치환하여 실행한다.

쉘입력독촉상태(shell prompt)에서 아래와 같이 alias를 입력하면 현재 체계에 정의된 alias의 목록을 볼수 있다.

이전에 alias관련명령을 실행해본적이 없다면 이것은 설치할 때부터 표준으로 설정된것이다.

[주의]

alias에는 메타문자를 사용할수 있으나 례외적으로 《=》는 사용할수 없다. 그것은 alias와 원래의 명령문을 대응시킬 때에 《=》를 사용하기때문이다.

아래와 같이 여러 명령들에 대해 alias로 예약해줄수 있다.

```
kcc@linux:~
# alias
alias cp='cp -i'
alias l.='ls -d .[a-zA-Z]* —color=tty'
alias ll='ls -l —color=tty'
alias ls='ls —color=tty'
alias mv='mv -i'
alias rm='rm -i'
alias vi='vim'
```

원래 cp명령은 복사받을 파일이 이미 존재하는가에 구애됨이 없이 복사하지만 cp명령에 i설정값을 첨부하면 같은 이름의 파일이 존재할 때에 덧쓰기 여부를 문의한다. 이 설정값의 사용은 실수로 하여 덧쓰기되는것을 피할수 있게 하므로 미리 cp명령에 i설정값까지 주어 cp별명(우의 화면출력내용에서 《cp-i》)을 지정하였다. 이와 같은 명령은 대부분이 표준으로 설정된 내용이다. 그러면 cp별명을 해제하여보자. alias의 해제명령은 unalias이다.

```
# unalias cp
```

alias명령으로 예약된 내용을 확인해보자.

```
kcc@linux:~
# alias
alias l.='ls-d .[a-zA-Z]* —color=tty'
alias ll='ls -l —color=tty'
alias ls='ls —color=tty'
alias mv='mv -i'
alias rm='rm -i'
alias vi='vim'
```

우의 화면에서 보면 지정해주었던 cp별명이 없어졌다. 다음과 같이 다시 cp별명을 생성할수 있다.

```
# unalias cp='cp-i'
```

다시 alias명령으로 확인하면 생성된 cp별명을 확인할수 있다. 우와 같은 방법으로 자주 사용하는 명령문과 설정값들을 간단하게 줄여 사용할수 있다.

실례로 많이 사용하는 tar명령과 설정값을 다음과 같이 만들어놓으면 편리하다.

```
# alias tarx='tar xvpf'
# alias tarc='tar cvpf'
# alias tarz='tar xvpfz'
```

다른 명령문도 같은 방식으로 적용할수 있다. alias를 설정한 상태로 원래의 명령문을 사용하기 위해서는 다음과 같이 입력해준다.

```
# \cp
```

또는

```
# /bin/cp
```

명령문앞에 《\》(역사선)이 붙어있으면 별명(alias)지정을 무시하고 원래의 명령을 실행하라는 의미이다. 또한 절대경로를 입력하여 명령문을 실행하는 경우에는 alias를 적용할수 없다.

【관련명령문】

unalias: alias를 해제한다.

apm

usr/bin/apm

학습장형콤퓨터의 전원관리

학습장형콤퓨터와 같은 장치에서 전지(Battery)의 현재 여유량을 확인하여 상태에 따라 체계를 대기 및 완료시킨다.

형식 : apm[-설정값]

-V, —version : apm프로그람의 판본(Version)을 보여준다.
-v, —verbose : APM BIOS의 판본과 전원상태에 관한 정보를 보여준다.
-m, —minuter : 전지의 남은 시간을 보여준다(단위: 분)
-s, —suspend : 체계를 정지상태로 만든다.
-S, —standby : 체계를 대기상태로 만든다.
-I : AC전원을 사용할 때에 APM의 체계정지, 대기지령을 무시한다. 학습장형콤퓨터의 경우에 전지를 사용하지 않고 AC전원을 사용할 때에 리용된다.
-n : AC전원을 사용할 때에 APM의 체계정지, 대기지령을 무시하지 않도록 한다.

학습장형콤퓨터사용자의 경우에 전지의 관리는 아주 중요한 문제로 되는데 apm명령문은 APM(Advanced Power Management) Bios의 정보가 들어있는 /proc/apm을 읽어들여 체계의 전원상태를 보여준다. 이 명령은 전지의 현재 여유량을 알수 있게 하며 체계를 정지상태, 대기상태로 만들수 있게 한다. 그러므로 학습장형콤퓨터상에서의 Linux사용자들에게는 필수적인 명령이다.

그러면 아래와 같은 방법으로 전지의 남은 시간을 확인하여보자.

```
# apm -m
5
```

여기서 5는 현재의 상태에서 계속 사용하려는 경우 앞으로 사용할수 있는 전지의 사용시간을 분으로 나타낸것이다. 탁상형콤퓨터나 학습장형콤퓨터를 직접 전원에 련결한 상태에서 이 명령을 실행하면 다음과 같은 통보문이 나온다.

```
AC on-line, no system battery
```

【관련명령문】

amp : 전원관리데몬

battstat_applet : 전지의 현재 여유량을 실시간적으로 확인할수 있게 하는 애플리트프로그람

참 고

학습장형콤퓨터와 Linux

최근에는 학습장형콤퓨터에 최적화된 특정한 설치를 선택적용할수 있게 되면서 학습장형콤퓨터에 Linux를 설치하는것이 그리 어렵지 않은것으로 되였지만 여전히 일반적인 설치보다는 품이 든다.

최근의 Linux 2.4.X판핵심부(Kernel)에서는 USB를 비롯하여 다양한 장치들을 지원하고있지만 종전의 2.2X판핵심부에서는 학습장형콤퓨터용의 하드웨어를 적지 않게 지원하지 못하였다. 이것은 일반적인 탁상형콤퓨터보다 작고 가벼우며 이동성이 편리해야 하는 학습장형콤퓨터의 특성으로부터 부단히 갱신발전되여 나오는 최신의 관련장치를 Linux가 시기적절하게 지원하지 못한데 있다.

아직도 최근에 나온 하드웨어나 새로운 표준에 준하고있는 하드웨어를 일부 지원하지 못하는 경우가 있지만 대부분의 학습장형콤퓨터들에는 탁상형콤퓨터에서와 같이 Linux를 설치하여 사용할수 있다.

appletviwer /usr/java/bin/appletviwer

Java애플리트의 실행

웨브페지를 읽어 Java애플리트를 실행하고 오유수정한다.

형식 : /usr/java/bin/appletviewer[설정값]URL

URL: 동작할 Java 애플리트가 있는 주소 또는 경로
-debug : Java debugger(jdb)에서 애플리트를 실행한다.
-encoding name : 입력되는 HTML 파일부호화를 지정
-J opt : Java 명령에 사용하는 설정값을 련결한다.

appletviewer는 인터네트상의 Java애플리트나 사용자자신의 콤퓨터상에 있는 Java애플리트를 따로 보기 위한것으로서 주로 애플리트가 정상동작하는 가를 확인하고 오유수정하기 위해 사용된다. 애플리트를 번역(compile)하여 확인할 때마다 웨브열람기를 사용하는것은 효률적인것이 못된다. 이때에 이 명령문을 사용하여 손쉽게 확인할수 있다.

appletviewer는 본래 UNIX계렬의 한 조작체계인 Solaris에서 사용하던 명령이였지만 지금은 Linux에도 jdk를 설치하면 사용할수 있다. 애플리트 그 자체는 불러들일수 없으며 애플리트가 포함된 html페지를 읽어 수행한다.

[주의]
JDK란 Sun에서 제공하는 Java개발환경으로서 여기에는 Java번역기, Java가상기계(JVM), 오유수정프로그람(Debugger) 등 응용프로그람개발을 위한 도구들이 포함되여있다.

[주의]
jdk를 설치하였는데도 appletviewer명령을 사용할수 없는 경우에는 환경변수 PATH를 확인한다.

애플리트는 단독으로 실행할수 없다. 사용할 때에는 html파일안에 다음과 같은 내용을 삽입하여 불러들여야 한다.

```
<object width="pixelWidth" height="pixelHeight">
    <param name="code" value="MyClass.class">
    <param name="object" value="serializedObjectOrJava Bean">
    <param name="codebase" value="classFileDirectory">

    …
    (기타 필요한 내용)
</object>
```

직접 만든 애플리트인 MyClass를 시험해보려는 경우에 먼저 우와 같은 내용으로 MyClass.html을 보관한다. 그다음 MyClass.html을 불러들여 애플리트를 확인한다.

```
# appletviewer MyClass.html
```

Java애플리트가 있는 웨브페지를 오유수정하기 위해 불러들일 때에는 아래와 같이 웨브페지주소와 debug설정값을 함께 사용한다.

```
# appletviewer MyClass.html
```

```
# appletviewer -debug http://www.kcctongbo.co.kp
```

참 고

애플리트(Applet)

애플리트는 소규모의 동화상이나 간단한 기능을 처리하는 작은 프로그람으로서 주로 Java언어와 같은 오브젝트지향언어로 만들어진다. 이것은 독립적으로가 아니라 다른 프로그람에 덧붙어 동작하는것이 일반적이다.

Linux에서 KDE나 Gnome에서 사용하는 간단한 기능의 프로그람도 애플리트라고 불리운다. WWW가 널리 사용되기 이전에는 Windows체계와 함께 기본으로 제공되던 작은 프로그람들, 례하면 기록장(memo)이나 그림그리기프로그람(paint) 등을 《애플리트》라고 부르기도 하였지만 현재 이러한 표현은 거의나 사용하지 않는다.

apropos /urs/bin/apropos

관련명령의 찾기

whatis DB를 검색하여 검색하는 명령문과 관련이 있는 명령문을 간단한 설명과 함께 보여준다.

형식 : apropos [실마리어]

실마리어(keyword) : 검색하려는 문자렬

apropos는 whatis DB를 검색하여 관련명령문과 그 명령문에 대한 간단한 설명을 보여준다.

whatis는 whatis DB에서 명령문만 검색하여 출력하지만 apropos는 명령문과 설명문 모두를 검색하여 결과를 보여준다.

실례로 아래와 같은 정황에서 이 명령문을 사용할수 있다.

Linux에서는 gimp와 같은 도형처리도구를 사용하지 않고도 bitmap파일을 jpeg형식으로 압축할수 있다. 그런데 례를 들어 그 명령이 생각나지 않는 경우에 다음과 같이 입력한다.

```
kcc@linux:~

# apropos jpeg
cjpeg      (1) –compress an image file to a
                JPEG file
djpeg      (1) –decompress a JPEG file to an
                Image file
Jpegtopnm  (1) –convert JPEG/JFIF file to protable
                pixmap or graymap
jpegtran   (1) –lossless transformation of JPEG
                files
ppmtojpeg  (1) –convert portable pixmap into a
                JPEG/JFIF file
rdjpgcom   (1) –display text comments from a JPEG
                file
wrjpgcom   (1) –insert  text comments into a JPEG
                file
```

[주의]

GIMP는 GUN Image Manipulation Program의 략자로서 피터 매티스(Peter Mattis)와 스펜서 킴볼(Spencer Kimball)이 개발한 UNIX용의 2D도형처리프로그람이다.

이 명령을 사용하면 명령이나 기본설명문에 jpeg라는 단어가 들어가는 명령을 찾아볼수 있다. 이 실례에서 찾아보려는 명령문인 일반파일을 jpg파일로 만들어주는 명령문은 제일 우에 있는 cjpeg이라는것을 확인할수 있다. 또한 jpeg파일을 bmp나 그외의 파일로 만들어주는 명령문은 djpeg라는것도 확인할수 있다.

【관련명령문】

man : -k 설정값을 사용하면 apropos 와 같은 기능을 한다.

whatis : whatis DB 에서 명령문과 일치하는 요약내용을 검색하여 보여준다.

참 고

whatis DB

명령문과 명령문에 대한 기본설명을 모아둔 파일이다. /var/cache/man/에 whatis라는 행별로 구분되여있는 본문자료기지파일로 존재한다. appros와 whatis 명령은 이 파일에서 해당 명령을 검색하여 내용을 보여준다. Whatis DB는 makewhatis명령으로 파일을 생성하거나 갱신할수 있다.

whatis DB갱신

makewhatis는 체계에 등록된 최신 도움말문서페지(man페지)를 검색하여 whatis DB를 갱신한다.

```
#/usr/sbin/makewhatis -u
```

arch

/bin/arch

CPU형의 보기

체계의 CPU형을 보여주는 명령문이다.

형식 : arch

arch명령문은 체계의 CPU형을 보여준다. 체계의 CPU형은 영문으로 《system architecture》라고도 표기하므로 architecture의 맨 앞의 4문자를 따서 이 명령으로 서술되였다. x86의 Intel계렬 CPU인 경우에 I386, I486, I586, I686, athlon 등으로 보여주며 기타 CPU형에 한해서는 alpha, sparc, arm, m68k, mips, ppc 등으로 보여준다.

```
kcc@linux:~

# arch
I686
```

참 고

Intel계렬 CPU

CPU는 여러 제작업체들로부터 많은 종류가 나오고있는데 그가운데서 현재 가장 널리 쓰이고있는 Intel계렬의 CPU에 대한 간단한 설명을 아래에 주었다.

4004(1971년)	Intel회사가 처음으로 만들어 내놓은 4bit CPU로서 기본적인 산수연산만을 할수 있다.
8008(1972년)	4004의 8bit판으로서 0.2MHz로 동작한다.
8080(1974년)	첫 조작체계인 CP/M을 사용한 Z80 CPU의 원형제품이 였다.
8086(1978년)	최초의 16bit 자료모선을 사용한것으로서 IBM PC-XT로서 널리 알려졌다. 4.77, 8, 10MHz속도로 동작한다.
8088(1979년)	8086과 같은 16bit등록기를 사용하지만 1970년대 후반기에 하드웨어와의 호환성을 유지하기 위해 외부

	모선을 8bit로 다시 설계한 제품이다. 4.77, 8MHz로 동작한다.
80186/80188 (1979년)	8086과 8088에 토대하여 내부의 마이크로코드를 개선하고 CPU의 걸모양을 PGA의 형태로 만든 제품이다.
80286 (1979년)	16bit자료모선, 24bit주소화, 16bit등록기 등을 받아들였다. 6MHz로 동작하지만 4.77MHz인 8088처리기에 비해 4배성능을 낼수 있으며 8, 10, 12MHz의 속도로 동작하는것도 있다. IBM PC-AT로 널리 알려졌다.
80386(DX/SX /SL)(1985년)	최초의 32bit Intel CPU이다. 실방식(real mode), 보호방식(protected mode)과 함께 virture 8086방식(여러개의 가상8086방식을 실행시킬수 있음)을 받아들여 가장 처음으로 다중과제처리를 실현하였다. 일반적으로 386, i386 이라고도 한다.
80486(DX/SX/DX2 /DX4/(1989년)	80386에 L1 cache, clock mutiple, 3.3V지원 등을 추가하였으며 현재의 Pentium CPU의 설계원형으로 되였다.

i386

rpm파일을 설치할 때에 《i386》이라는 표기를 자주 보게 된다.

이것은 이 rpm파일이 80386에 대해 아래준위호환성을 가지는 대부분의 콤퓨터들에서 원만히 설치될수 있도록 80386환경에서 만들어졌다는 의미로서 rpm파일을 묶을 때에 표준으로 되여있다(i686으로 하기 위해서는 설정값을 주어야 한다). 이 경우 Intel계렬 CPU를 사용하는 대부분의 콤퓨터에서 원만히 동작하는 우점이 있지만 Pentium이후의 환경에 최적화된것은 아니다.

arp /sbin/arp

체계의 MAC주소확인

련결하려고 하는 체계의 MAC주소확인명령문

형식 : arp[설정값]

-v : arp상태를 보여준다.
-t type : ARP cache에 올라있는 형식을 검색한다. ether(Ethernet) 또는 ax25(AX.25 packet radio) 등이 있으며 ether가 기본형식이다.
-a [hosts]: 등록된 호스트중 지정한 호스트의 내용을 보여준다. 호스트를 지정하지 않으면 등록된 모든 호스트를 보여준다.
-d [host] : 지정한 호스트를 목록에서 삭제한다.
-s host hardware-address: 호스트의 장치주소 즉 호스트 MAC주소를 추가한다. 이써네트기판의 경우에는 6자리의 16진수로 되여있다.
-f file: 파일에 있는 목록을 추가한다.

이것은 TCP/IP명령문이다. 체계사이의 통신을 위해서는 상대방의 MAC주소가 필요한데 이때에 arp는 ARP(Address Resolution Protocol)를 리용하여 련결하려고 하는 체계IP에 신호를 보내여 MAC주소를 받아온다.

부분망의 arp정보는 련결효률을 높이기 위해 /proc/net/arp에 보관된다.

cache에 보관된 정보는 추가 및 삭제할수 있다. 이와 같이 보관된 arp cache의 내용을 자세히 보려는 경우에는 다음과 같이 실행한다.

```
kcc@linux:~

# arp -v
Address                 HWtype   Hwaddress           FlagsMask   Iface
ebook.kcctongbo.co.kp   ether    00:04:76:71:F4:88   C           eth0
211.40.61.1             ether    00:00:0C:76:BF:6C   C           eth0
211.40.61.36            ether    00:50:DA:90:44:2A   C           eth0
211.40.61.211           ether    00:04:76:72:3B:85   C           eth0
Entries:4               Skipped:0   Found:4
```

ebook.kcctongbo.co.kp라는 호스트에 대한 정보만 알려고 하는 경우에는 다음과 같이 한다.

```
kcc@linux:~
# arp -a ebook
ebook.kcctongbo.co.kp(211.40.61.108) at
00:04:76:71:F4:88 [ether] on ethe0
```

해당 체계에서 이전의 이써네트기판대신에 새로운 이써네트기판을 설치했을 때에 내부망련결이 잘 되지 않는 경우가 있다. 이것은 arp cache에 이전 이써네트기판의 MAC주소가 보관되여있으므로 같은 IP를 사용하여도 인식하지 못하는 경우이다. 이때에는 arp -d명령을 사용하여 정보를 삭제한 후에 다시 시도한다.

```
# arp -d ebook: 호스트이름이 ebook인 정보를 삭제
```

```
kcc@linux:~
# arp -v
Address               HWtype     Hwaddress            FlagsMask     Iface
ebook.kcctongbo.co.kp                                 (incomplete)  eth0
211.40.61.1           ether      00:00:0C:76:BF:6C    C             eth0
211.40.61.36          ether      00:50:DA:90:44:2A    C             eth0
211.40.61.211         ether      00:04:76:72:3B:85    C             eth0
Entries:4             Skipped:0  Found:4
```

【관련명령문】

arping : 대상주소에 ARP파케트를 전송한다.

Arpwatch : Ethernet/IP주소의 수행경로를 추적한다.

arpsnmp : Ethernet/IP주소의 수행경로를 추적한다.

tcpdump : 망대면부에서의 파케트머리부들을 출력한다.

참 고

MAC주소 (MAC address)

망기판제작회사들에서 기판마다에 부여한 48bit의 고유번호이다. IP주소만으로 망운영중에 장애가 발생하는 경우에는 문제발생시에 체계를 구분할수 있는 정보가 부족하여 그에 대처하기가 곤난하다. 간단한 례로 IP충돌이 발생한 경우에 그것을 알수 있는 다른 정보가 필요한데 이로부터 생성되는것이 MAC주소이다. 두 체계의 IP를 같게 하여 문제가 발생한 경우에도 매 체계는 이 망기판의 고유번호로 구분할수 있다.

ARP

Address Resolution Protocol의 략자로서 IP망우에서 IP주소를 MAC주소와 대응하기 위해 사용하는 통신규약이다. 례를 들어 IP호스트 A가 IP호스트 B에게 IP파케트를 전송하려고 할 때에 IP호스트 B의 MAC주소를 모르는 경우. ARP규약을 사용하여 목적지 IP주소 B와 Broadcasting MAC주소 FFFFFFFFFFFF를 가지는 ARP파케트를 망전송한다. IP호스트 B는 자기의 IP주소가 목적지의 ARP파케트를 수신하면 자기의 MAC주소를 A에게 돌려보낸다. 이와 같은 방식으로 수집된 IP주소와 이에 해당하는 MAC주소정보는 매 IP호스트의 《ARP cache》라는 주기억령역에 표(Table)형태로 보관하여 다음의 파케트를 전송할 때에 다시 사용할수 있다. ARP와는 반대로 IP호스트가 자기의 물리망주소는 알지만 IP주소를 모르는 경우에 봉사기로부터 IP주소를 요청하기 위해서는 RARP(ReverseARP)를 사용한다.

통신규약(Protocol)

통신규약은 정보기구들사이, 즉 콤퓨터호상간 또는 콤퓨터와 말단기구사이에 정보를 교환할 때에 필요로 하는 규약이다. 통신규약이란 련결된 호상간의 접속이나 단절, 통신방식, 주고 받을 자료의 형식, 오유검출방식, 코드변환방식, 전송속도등에 관하여 정하는것으로서 기종이 다른 콤퓨터들간의 호상통신을 위해서는 표준통신규약을 사용하여 통신망을 구축해야 한다. 이와 같은 표준통신규약가운데서 대표적인것이 인터네트에서 널리 사용되고있는 TCP/IP이다.

arping

/sbin/arping

망련결확인

arp요청을 리용한 망련결확인명령문.

형식 : arping[설정값]

-c count : arping의 개수를 지정한다.
-w timeout : w설정값뒤에 시간(초)을 지정하여 그 시간(초)까지만 apring를 실행한다.
-i device : 이써네트장치를 지정한다.
-s source : 원천(source) IP를 지정한다.

ping명령의 변종으로서 ICMP파케트대신에 ARP요청과 응답으로 사용중인 IP주소로 MAC(Meduim Access Control)주소를 찾거나 MAC주소가 있을 때에 IP주소를 찾을수도 있다. 또한 ICMP파케트를 차단한 호스트라도 다만 련결성만을 검사하려는 경우에 arping은 효률적인 수단으로 된다.

아래에 자신의 체계로 arping을 3번 보내여 확인하는 실례를 주었다.

```
kcc@linux:~

# arping -c3 localhost
ARPING  127.0.0.1  from  211.155.153.58  etho
Unicast  reply from  127.0.0.1  [00:01:96:72:47:C3]  2.101ms
Unicast  reply from  127.0.0.1  [00:01:96:72:47:C3]  2.025ms
Unicast  reply from  127.0.0.1  [00:01:96:72:47:C3]  2.049ms
Sent  3  probes (1 broadcast(s))
Received  3  response(s)
```

체계에 여러개의 망기구가 장치되여있어 매 장치별 련결성을 검사하려는 경우에 《-I》설정값을 리용하여 장치를 지정하여준다.

```
kcc@linux:~

# arping -I eth1 -c3 www.linuxroot.co.kp
APPING 211.255.252.57 from 211.255.253.58 eth0
Unicast reply from 211.255.252.57 [00:01:96:72:47:C3] 2.231ms
Unicast reply from 211.255.252.57 [00:01:96:72:47:C3] 2.245ms
Unicast reply from 211.255.252.57 [00:01:96:72:47:C3] 2.119ms
Sent 3 probes (1 broadcat(s))
Received 3 response(s)
```

【관련명령문】

ping : 망련결성을 분석한다.

참 고

ICMP(Internet Control Message Protocol)

ICMP는 호스트봉사기와 인터네트게트웨이사이에서 통보문(Message)을 조종하고 오유를 알려주는 통신규약이다. 경로기가 망경로를 설정할수 없거나 자료를 전송할수 없는 때에 망우에 존재하는 이러한 문제를 해결하도록 정보를 전송하는데 필요한 통신규약이다. ping 명령도 이 통신규약을 사용한다. MAC의 설명에 대해서는 arp를 참고할것.

TCP(Transmission Control Protocol)

TCP는 인터네트상의 콤퓨터들사이에서 자료를 통보문의 형태로 전송하기 위해 IP와 함께 사용되는 통신규약으로서 OSI 통신모형의 4계층인 전송층에 속한다. TCP는 통보문들이 매 단의 응용프로그람사이에서 교환되는동안 련결유지되는것으로 하여 련결지향통신규약으로 알려져있으며 자료파케트의 순서를 제공한다.

UDP(User Datagram Protocol)

UDP는 IP를 사용하는 망내에서 콤퓨터들사이에 통보문들이 교환될 때에 제한된 봉사만을 제공하는 통신규약이다. TCP와는 달리 도착하는 자료파케트들의 순서를 제공하지 않는다. 그대신에 처리속도가 빠르므로 교환해야 할 자료가 매우 적은 경우, TCP보다 효률적으로 사용할수 있다.

at

/usr/bin/at

작업예약

명령문이나 스크립트의 실행을 예약한다.

형식 : at[설정값]시간 [날자][+증가시간]

시 간 : 명령문이나 스크립트를 실행할 시간을 지정한다.
+증가시간 : 증가시간만큼 지난후에 실행한다.
-q queue : queue(대기렬)이름을 지정한다. 대기렬이름으로는 a~z, A~Z를 사용할수 있다. 대기렬이름을 지정하지 않으면 at는 a를, batch는 b를 사용한다. 영어자모순서로 뒤의 이름을 가지는 대기렬은 보다 큰 niceness값을 가진다. =queue는 현재 수행되는 작업에 대한 대기렬로 예약되여있다. 대문자이름의 대기렬에 추가된 작업은 batch의 작업처럼 처리된다.
-m : 출력결과가 없는 경우에도 작업이 완료될 때 사용자에게 우편(mail)을 보낸다.
-f 파일명 : 스크립트파일을 실행하여줄 때 사용한다.
-l : 예약된 작업목록을 보여준다.
-v : 작업이 수행될 시간을 보여준다.
-d : 예약된 작업을 삭제한다(linux).
-r : 예약된 작업을 삭제한다(Unix).

at는 등록된 명령문을 요구하는 시간에 실행하게 하는 기능을 한다. 즉 체계부하가 적은 시간에 실행시키거나 예약된 시간에 꼭 실행되여야 하는 작업을 자동으로 처리하려고 할 때에 사용한다.

[주의]

at명령을 실행하려면 atd데몬이 동작하고 있어야 한다.

at에 의해 실행할 시간의 설정

예약작업기능이므로 시간에 대한 설정이 중요하다. at의 시간설정에는 날자와 시간을 정해주는 시간설정, 현재시간설정, 정해진 시간으로부터의 증가시간으로 설정 등의 방법이 있다.

시간표현방법

분 류	형 식	설 명	례
시 간	hh:mm	hh(시간), mm(분)으로 설정.	16:18
	am, pm	표시가 없는 경우에 Linux나 Unix에서는 하루를 24시간으로 표현한다. 오전과 오후를 의미하는 am과 pm을 리용하여 12시간단위로 표현할수 있다.	4:18pm
	midnight noon teatime now	midnight(00:00), noon(12:00), teatime(16:00) 같은 서술형시간으로도 지정할수 있다. now는 현재시간이며 증가시간설정에 많이 쓰인다.	
날 자	yyy-mm-dd	일반적인 날자구성으로 표현..	2004-4-13
	month num	례하면 4월 13일의 경우에 《April 13》으로 표현하며 혹은 략자로 《Apr 13》으로 표현할수도 있다.	April 13
today tomorrow		오늘 또는 래일로 지정할수 있다. 매 요일별로도 표현할수 있다.	

[주의]

한개의 명령뿐만아니라 여러개의 명령을 등록하여 사용할수도 있다.

test1파일을 만들어 주는 명령문인 《touch test1》이라는 명령문을 at명령으로 예약하여 례를 들어 2004년 4월 13일 0시에 실행하도록 하는 경우를 가상해보자. 2004년 4월 13일 0시는 00:00 2004-04-13 또는 《midnight April 13》 등으로 시간을 표현해줄수 있다.

```
kcc@linux:~

# at 00:00 2004-04-13
at>touch test 1
at><ctrl+d>
```

예약된 작업확인

```
kcc@linux:~
# at -l
20 2004-04-13 00:00 a root
```

at에 예약된 작업정보

작업번호	날 자	시간대기열	이 름	사용자
20	2004-04-13	00:00	a	root

- 작업번호 : at에 등록된 작업번호, 등록된 순서로 할당되며 at -d, 또는 atrm을 리용하여 at작업을 삭제하여줄 때에 사용한다.
- 날자 및 시간 : 등록된 명령이 실행될 시간이다.
- queue이름 : 《-q》를 리용하여 queue이름을 지정한다.
- 사용자 : at에 작업을 예약한 사용자.

증가시간을 리용한 설정

현재 시간으로부터 6시간후에 Control.sh스크립트가 실행되도록 예약하려는 경우에는 다음과 같이 f설정값을 사용하여 예약작업으로 등록한다.

또한 《<》을 리용하여도 같은 결과를 얻을수 있다.

```
# at now+6 hours -f Control.sh: 파일을 예약작업으로 등록
```

```
# at now+6 hours <  Control.sh: <을 리용한 예약작업등록
```

예약한 작업을 확인해보자.

```
kcc@linux:~
# at -l : 예약된 작업의 확인
20 2004-04-13 00:00 a root
21 2004-04-12 18:00 a root
```

20번으로 예약된 작업을 삭제하려는 경우에는 아래와 같이 삭제할 작업번호를 지정해준다.

```
kcc@linux:~

# at -d 20

# at -l
21 2004-04-12 18:00 a root
```

20번에 예약된 작업이 취소된것을 확인할수 있다.

at의 사용권한 설정

at명령문은 at.allow와 at.deny의 두 파일을 리용하여 일반사용자의 at 명령문사용권한을 지정해줄수 있다.

/etc/at.allow파일이 있으면 at.allow에 기록된 사용자만이 at명령문을 사용가능하며 /etc/at.allow 파일이 없으면 /etc/at.deny에 기록되지 않은 사용자만 사용가능하다. 두 파일이 모두 없는 경우에는 관리자만이 사용할수 있다.

【관련명령문】

atq : 예약된 작업목록을 보여준다.
atrm : 예약된 작업목록을 삭제한다.
batch : 체계부하가 지정된 값이하인 때에 명령을 실행한다.
cron : 정기적으로 예약된 작업을 수행한다.

참　고

at와 cron

at와 cron은 작업의 예약실행을 가능하게 한다. at는 특정시간을 정하기가 쉬운것으로 하여 비정기적인 시간을 예약하여 사용하기에 편리하며 정기적으로 수행해야 하는 작업인 경우에는 cron을 사용하는것이 더 편리하다. cron은 월단위, 주단위, 일단위, 시간단위로 정기적인 시간설정이 가능하다.

	at	cron
공통점	작업을 예약하여 실행한다.	
용 도	비정기적인 작업예약	정기적인 작업예약
관련데몬	atd	crond
사용권한	/etc/at.allow허가 /etc/cron.allow허가	/etc/at.deny거부 /etc/cron.deny거부

UNIX시간 (Unix time)

date +%s 명령을 사용하여 확인할수 있다. UNIX의 time() 함수는 1970년 1월1일 0시부터 현재까지 경과된 시간을 초단위의 옹근수값으로 반환하여 시간을 계산하고있다. time_t 변수는 1970년 1월 1일 0시부터의 초시계를 옹근수로 나타낸다. 이 점을 리용하여 파일이름이 중복되지 않도록 만들기 위해 UNIX시간을 리용하여 날자도장(Time Stamp)을 파일이름에 붙이는 경우도 많다. 그러나 문제가 있는바 그것은 UNIX날자값이 4Byte공간으로 제한되여있어 이 초시계를 32bit의 옹근수값으로 표현하는 현 상태에서는 2038년까지의 시간만을 계산할수 있다는것이다. 물론 아직 시간적여유가 있으므로 앞으로 이 문제는 해결될수 있을것이다.

atq

/usr/bin/atq

예약작업보기

at를 사용하여 예약한 작업목록을 보여준다.

형식 : atq[설정값]

[Linux]
-q queue : 지정된 대기렬에 예약된 작업만 보여준다.
-v : 판본을 보여준다.

[Unix]
-c : 예약된 작업을 시간별로 정렬하여 보여준다.
-n : 대기렬에 예약된 모든 작업개수를 보여준다.

at에 의해 예약된 작업목록을 보여준다. at -l와 같은 기능을 한다.

이 가운데서 특정대기렬에 예약된 작업을 보기 위해서는 atq -q라고 입력한다. 아래에 t대기렬에 예약된 작업을 보여준다.

```
kcc@linux:~
# atq -q t
27     2004-04-16 10:00 t root
```

【관련명령문】

at : 작업을 특정시간에 예약한다.
atrm: 예약된 작업목록을 삭제한다.

atrm

/usr/bin/atrm

예약작업삭제

at를 사용하여 예약한 작업중 불필요한 내용을 삭제한다.

형식 : atrm 작업번호

작업번호 : atq를 사용하여 확인할수 있다.

atq를 사용하여 예약된 작업목록을 확인한 다음 첫번째 항목에 있는 작업번호를 확인하자.

```
kcc@linux:~

# atq
29      2004-04-13 12:00 A root
25      2004-04-14 05:28 A root
27      2004-04-16 10:00 t  root
28      2004-04-15 12:00 A root
```

우에서 2002년 4월 16일 10시에 예약된 작업을 삭제해보자.
작업번호가 27이므로 다음과 같이 삭제한다.

```
# atrm 27
```

제대로 삭제되였는가를 확인해보자.

```
kcc@linux:~

# atq
29      2004-04-13 12:00 A root
25      2004-04-14 05:28 A root
28      2004-04-15 12:00 A root
```

27번 작업이 목록에서 사라졌다.

【관련명령문】

at : 작업을 특정시간에 예약한다.
atq : 예약된 작업목록을 보여준다.
batch : 체계부하가 기정값이하인 때에 명령을 실행한다.
cron : 정기적으로 예약된 작업을 실행한다.

참 고

예약작업시간의 결정

작업의 예약실행은 일반적으로 체계과부하를 피할 목적으로 한가한 시간에 작업을 실행하기 위해 적용하는데 일반적으로 밤시간에 설정하는 경우가 많다.

그러나 여러 사용자가 예약작업을 설정한 경우에 예약시간이 새벽 3시나 4시로 정해지는 때도 많은데 이와 같이 야간이라도 특정시간에 부하가 몰리는 경우에는 체계에 문제를 발생시킬수 있다. 그러므로 예약작업시간을 결정할 때에는 새벽 3시 36분 혹은 4시 11분 등과 같이 분까지 설정하여 같은 시간에 작업이 집중되는것을 피하도록 해야 한다.

awk /bin/awk

패턴처리언어

원본문서에서 패턴을 검사하여 요구하는 값을 얻는다.

형식 : awk[설정값] -F 'script' [변수=값][파일] 또는
awk[설정값] -f 스크립트파일 [변수=값][파일]

스크립트파일 : awk스크립트로 작성된 파일.
변수 : awk내의 변수를 지정한다.
파일 : 대상파일
-F : 식별자를 나타낸다. -F로 식별자를 지정하지 않는 경우, 공백을 식별자로 사용한다.
-f : 스크립트파일을 리용하는 경우에 사용한다.

표준입력으로 값을 받아 awk스크립트를 통해 요구하는 표준출력을 얻어낸다. awk는 일정한 규칙을 가지고있는 자료를 처리하여 계산, 통계, 비교분석, 자료추출 등에 다양하게 사용될수 있다.

sed와 비슷한 기능을 하며 awk와 sed의 우점을 모아 만들어진 perl로 발전되였다. 단순한 구조를 가진 자료에서 보다 효과적으로 사용될수 있으므로 지금도 많이 사용되고있다.

awk는 편리하고 효률적으로 사용될수 있다. 단순히 명령문으로 사용되거나 다른 스크립트에 sed와 함께 다른 스크립트에 리용될수 있으며 awk만의 스크립트파일작성도 가능하다.

명령문의 사용

현재 등록부의 파일정보를 보기 위해 ls -al명령문을 리용하여 파일목록을 출력한다. 이 목록에서 파일의 권한과 파일이름만을 보려고 할 때에는 awk를 리용하여 손쉽게 편집할수 있다. ls -al명령으로 출력되는 자료를 awk가

파이프(|)로 받아 파일의 공백을 기준으로 나누어 첫번째 항목과 9번째 항목을 출력한다.

```
kcc@linux:~

# ls -al
drwxr-xr-x  21root  root  4096 6월 25 19:42     .
drwxr-xr-x  21root  root  4096 6월 25 19:42     ..
rw-r--r—    1 root  root  5782608 11월 20 2001  libqt-mt.so.23
rw-r--r—    1 root  root  5761641 11월 20 2001  libqt.so.2.3.2
rw-r--r—    1 root  root  33735 11월 20 2001    libqutil.so.1.0.0
-rw------   1 root  root  23376 5월 28 21:12    mbox
rwxr-xr-x   1 root  root  4017 5월 4 17:03      memorycheck
rw-r--r—    1 root  root  1698368 5 월 4 10:59  memoryinfo
```

```
kcc@linux:~

# ls -al | awk '{print $1, $9}'
합계
-rw-r--r--   libqt-mt.so.2.3
-rw-r--r--   libqt.so.2.3.2
-rw-r--r--   libqutill.so.1.0.0
-rw-------   mbox
-rwxr-xr-x   memorycheck
-rw-r--r--   memoryinfo
```

awk로 추출하여 파일의 권한과 파일이름만을 출력한다. 우와 같이 awk는 행을 받아 식별자를 통해 구분하고 print명령으로 출력하게 된다.

ls -al로 출력되는 행을 공백으로 구분하면 아래와 같이 구분된 $1~$9까지 항목이다. 매 항목은 공백을 기준으로 아래와 같은 번호로 구분된다. $0은 모든 항목을 의미한다.

-rw----	2	root	root	23376	5월	28	02:30	mbox
파일권한	하드련결번호	사용자	집단	용량	월	일	시간	파일이름
$1	$2	$3	$4	$5	$6	$7	$8	$9

파일목록내에서 1MB이상의 용량을 가지는 파일이름과 용량을 출력하려는 경우, awk의 조건문을 사용할수 있다.

1MB=1048576byte이므로 용량을 나타내는 5번째 항목값중에서 048576보다 큰 행을 출력하며 그 행안의 5번째와 9번째 항목값만을 출력한다.

```
kcc@linux:~
# ls-al | awk '$5 > 1048576{print $5, $9}'
5782608 libqt-mt.so.2.3
5761641 libqt.so.2.3.2
1698368 memoryinfo
```

awk를 리용하면 문서의 특정한 문자나 문자렬을 검색하여 나온 행만을 출력할수 있다.

```
kcc@linux:~
# awk /root/ /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root:bin,adm
adm:x:4:root,adm,daemon
disk:x:6:root
wheel:x:10:root
```

뿌리식별이름을 포함한 행만 검색하여 출력하였다.

첫번째 실례에서는 문자렬중 구분되여있는 항목을 지정하여 요구하는 항목만을 출력하였으며 두번째 실례에서는 요구하는 패턴을 가진 행만 검색하여 출력하였다.

이 두가지를 조합하면 요구하는 행의 특정한 항목만을 출력할수 있다.

두번째 실례에서는 /etc/group파일을 root라는 패턴으로 검색하여 root라는 문자렬을 포함한 행만이 출력되도록 하였다. 《:》표시로 구분해준 항목중 제일 먼저 나오는것이 《집단명》인데 다음 실례에서는 이 집단명만 출력되도록 하였다.

-F는 식별자를 정의해주는 설정값으로서 기본적으로는 공백을 사용하지만 아래의 실례에서와 같이 《-F:》로 식별자를 지정하여 사용할수도 있다.

```
kcc@linux:~
# awk -F: /root/'{print $1}' /etc/group
 root
 bin
 daemon
 sys
 adm
 disk
 wheel
```

/root/로 검색한 패턴부분에 정규표현식을 사용하면 보다 자세한 패턴을 검색할수 있다.

스크립트파일의 리용

시작(Begin)	시작단계로서 전체 스크립트를 위한 정의단계이다(Preprocessor).
실행(Routin)	실행단계로서 이 스크립트의 기능을 수행하는 단계라고 할수 있다.
끝(End)	마무리단계로서 결과를 출력한다.

awk의 스크립트구조는 시작, 실행, 끝 등의 3단계로 나누어져있다.

awk는 여러 연산자, 순환(loop)의 사용법 등이 C와 류사하므로 C언어를 습득한 사용자라면 어렵지 않게 사용할수 있다.

다음의 스크립트는 파일의 단어수를 검사한다. wc라는 명령문으로 쉽게 단어개수를 검사해볼수 있지만 여기에서는 간단한 awk스크립트실례에 의해 awk스크립트의 사용법을 보기로 한다.

awk스크립트	구 조	설 명
#!/bin/awk	선언	awk로 스크립트가 실행될수 있도록 선언한다.

BEGIN{
word=0:
}
	시작
	(Begin)
		변수 word를 0으로 초기화한다.

{
word+=NF;
}
	실행
	(Routin
	)
		NF는 매 행마다의 항목(field)수를 나타내는 awk 체계변수이다. awk는 식별자가 정의되어있지 않을 때에 공백을 식별자로 사용하므로 매 행의 단어수가 NF에 등록되며 《+=》연산자에 의해서 마지막 행의 단어개수까지 더해준다.

END {
print "Word Count:"
word;(End)
}
	끝
	(End)
		마무리단계로서 결과를 출력한다.

작성된 스크립트는 -f설정값으로 실행한다. wordcount.txt라는 임의의 본문문서를 방금 작성한 스크립트로 검사해보자. 작성된 awk스크립트파일을 실행하려면 awk -f [스크립트파일이름]형식으로 사용한다.

```
kcc@linux:~
# awk -f word.awk wordcount.txt
Word Count: 580
```

awk체계변수

변 수	내 용
$0	입력행의 모두
$n	입력행의 n번째 항목값
ARGC	명령행의 인수를 가지는 변수
ARGV	명령행의 인수를 포함하는 배렬
ENVIRON	환경변수들을 모아둔 관계형배렬
FILENAME	현재 파일이름
FS구	식별자를 정의. 공백을 기본으로 사용
FNR	입력파일의 기록총수(행수)
NF	현재 기록의 항목수
NR	현재 기록의 번호
OFMT	수자에 대한 출력형식
OFS	출력항목의 구분. 빈 행을 기본으로 사용
ORS	출력기록의 구분(newline을 기본으로 사용)
RLENGTH	지정한 패턴으로 검색되여 나온 문자렬의 길이
RS	입력기록의 구분(newline을 기본으로 사용)
RSTART	지정한 패턴으로 검색되여 나온 문자렬의 가장 앞부분

awk연산자

C언어를 참조하여 만들어졌으므로 사용법이나 종류에 있어서 C언어와 거의나 같다.

연 산 자	설 명
?	**조건연산** 사용자로 등록된 ID가 user1, user2, user3으로 되여있고 그중 검색하려는 내용이 1, 2, 3가운데서 어느것인가 명확치 않은 경우에 《?》를 리용하여 모두 검색할수 있다. #awk /user1?//etc/passwd user1:x:516:516::/home/user1:/bin/bash user2:x:517:517::/home/user2:/bin/bash user3:x;518:518::/home/user3:/bin/bash
||, &&, !	론리연산자인 론리합(or), 론리적(and), 론리부정(not)
~, !~	검색된 패턴에 부합되는것을 《참》으로 사용하려면 《~》을, 거짓으로 사용하려면 《!~》을 사용한다.
<, <=, >, >=, !=, =	비교연산자
+, -, *, /, %, ^	더하기, 덜기, 곱하기, 나누기, 나머지, 제곱
++, --	증가연산자, 감소연산자

참 고

표준입력과 표준출력

어떤 명령을 사용할 때에 파일이름이나 필요한 다른 요소를 함께 보여준다.
이 명령의 결과는 화면에 출력된다.

```
kcc@linux:~
# cat test.txt
이 파일은 표준입력과 표준출력을 설명하기 위한것이다.
표준입력은 test.txt 이다
```

우에서 cat라는 명령의 뒤에 test.txt라는 파일이름이 있는데 이것은 test.txt의 내용을 읽기하여 화면에 출력하라는 명령이다. 즉 test.txt는 원하는 작업을 하는데 필요한 입력요소이며 이 명령의 결과로서 test.txt의 내용인 두행의 문장이 화면에

출력되는것이다. 여기에서 test.txt와 같이 명령을 실행하기 위해 필요한 입력을 표준입력이라고 하며 명령의 실행결과를 표준출력이라고 한다. 또한 표준입력을 STDIN, 표준출력을 STDOUT로 표시하기도 하는데 이것은 Standard Input와 Standard Output의 략자이다. 표준입력을 받아 프로세스가 작업을 마치면 화면에 출력하는 명령행프로그람이 이 내용을 표준출력으로 내보낸다.

a2p

/usr/bin/a2p

awk-perl변환

awk스크립트를 perl스크립트로 바꾸는 명령문

형식 : a2p[설정값]파일명

-F〈문자〉: 식별자를 정의한다. awk의 항목을 나누는 기준으로 되는 식별자변수인 FS변수를 지정된 문자로 정의한다.
-o : 이전 awk형식의 스크립트를 변환한다.

a2p는 awk스크립트를 perl스크립트로 변환하는 명령문이다.
awk명령문에서 실례로 만든 간단한 스크립트를 변환해보자.
아래의 실례는 문서의 단어개수를 검사하는 스크립트이다.
word.txt라는 문서의 단어수를 검사하여 출력하도록 해보자.

```
kcc@linux:~

# awk -f word.awk word.txt
Word Count: 580
```

이 awk스크립트를 perl스크립트로 변환한다.

```
kcc@linux:~

# a2p word.awk

#!/usr/bin/perl
eval 'exec /usr/bin/perl -S $0 ${1+ "$@"}'
   if $running_under_some_shell;
                    #this emulates #! processing on NIH machines.
                    #(remove #! line above if indigestible)
eval '$', '$1', '$2;', while $ARGV[0]=~/^([A-Za-z_0-  9]+=)(*)/&& shift:
                    # process any FOO=bar switches
```

```
$[=1;    # set array base to 1
$,= ' ';    # set output field separator
$\="\n";    # set output record separator

$word=0;

while ( < > ) {
    chomp; # strip record separator
    @Fld=split(' ',  $_, 9999);

    $word+=$#Fld:
}

print  'Word Count: '. $word;
```

a2p는 perl스크립트로 변환하여 표준출력으로 변환된 문서를 화면에 보여준다. 이 표준출력을 word.perl이라는 perl스크립트파일로 만들어 실행하면 같은 결과를 얻을수 있다.

[주의]
표준출력에 대한 내용은 awk를 참고하시오.

```
kcc@linux:~
# a2p word.awk > word.perl
# perl word.perl word.txt
Word Count : 580
```

【관련명령문】

s2p : sed스크립트를 perl스크립트로 변환

B

badblocks /sbin/badblocks

불량블로크의 검사

기억장치의 불량블로크를 검사하는 명령문

형식 : badblocks[설정값][장치][블로크개수]

장 치: 검사하려는 장치를 지정한다.
블로크개수: 검사할 장치의 블로크개수

-b 블로크크기: 블로크크기를 byte수로 나타낸다.
-o 파일: 불량블로크의 검사내용을 파일에 보관한다.
-v : 불량블로크의 검사내용을 상세히 보여준다.
-w : 각 블로크에 몇가지 패턴으로 쓰기, 읽기를 하면서 검사한다.

[주의]
파일체계가 있는 기억장치에 -w를 사용하면 파일체계가 삭제되며 자료도 함께 손실되므로 자료가 기억되여있는 장치에 사용하여서는 안된다.

Badblocks는 매 디스크의 불량블로크를 검사하는 명령이다. 체계의 /dev/hda2디스크의 불량블로크를 검사한다. 이때에 블로크개수를 알기 위해 먼저 fdisk를 실행하여 디스크정보를 확인한다. 블로크개수는 다음과 같은 방법으로 확인할수 있다.

```
kcc@linux:~

# fdisk -l /dev/hda
Device      Boot  Start   End    Blooks    Id   System
/dev/hdal    *     1      261    2096451   83   Linux
/dev/hda2          262    587    2618595   83   Linux
/dev/hda3          588    848    2096482+  83   Linux
/dev/hda4          849    1027   1437817+  5    Extended
/dev/hda5          849    913     522081   83   Linux
/dev/hda6          914    1027    915673+  83   Linux swap
```

우의 명령결과로부터 /dev/hda2에는 2618595개의 블로크가 있다는것을 알수 있다.

```
kcc@linux:~

# badblocks -v /dev/hda2 2618595
Checking for bad blocks in read-only mode
From block 0 to 136521
136520 : 불량블로크가 발견. 디스크블로크
Pass completed, 1 bad blocks found.
1 개의 불량블로크가 발견되였다.
```

136521디스크블로크에서 1개의 불량블로크가 발견되였음을 보여준다.

참 고

불량블로크(Bad Block)

물리적으로 파괴된 디스크블로크를 말한다. 디스크에 블로크단위로 정보가 보관되는데 충격이나 로화로 인하여 불량블로크가 생길수 있다. 최근에 나오는 하드디스크들에서는 대부분 하드디스크가 자체로 불량블로크를 검사하여 관리한다.

Linux에서는 파일체계생성시에 불량블로크에 관한 정보를 초기화하여 가지게 되며 하드디스크의 로화로 인하여 불량블로크가 특정한 부분에 많은 경우에는 구획(Partition)을 나누어 특정부분의 사용을 금지하도록 한다.

블로크(Block)

블로크란 파일체계가 련속적으로 할당하는 자료의 크기를 의미한다. 례를 들어 파일체계의 블로크크기가 8KB라면 8KB까지의 파일크기는 디스크의 여러 부분에 나뉘여있지 않고 항상 물리적으로 같은 자리에 련속으로 존재한다. 이것은 디스크의 물리적블로크와는 또 다른것이다.

블로크의 크기가 크면 파일체계에 접근하는 성능은 좋아지지만 블로크 한개의 크기보다 작은 크기의 파일이 생성되는 경우에도 하나의 블로크를 랑비하게 하므로 디스크용량의 효률은 낮아진다. 반면에 블로크의 크기가 작으면 디스크용량의 랑비는 최소화할수 있으나 성능이 떨어지는 약점이 있다. UNIX나 Linux에서는 일반적으로 512byte크기의 블로크를 사용한다.

banner

/usr/games/banner

문자확대출력

이전 시기에 큰 문자의 출력을 위해 많이 사용된 명령이다.

형식 : banner[-wn]문자렬

> -w : 기본적으로 출력되는 칸수는 132칸이지만 -w설정값을 사용하여 렬수를 정해줄수 있다. n값이 없는 경우에 80렬로 표현된다.
>
> n : 확대되여 출력되는 렬수

문자렬을 확대출력할 때에 사용된다. 문자가 #으로 표현되여 확대출력된다. 현재는 거의나 사용하지 않는 명령이지만 간혹 ASCII Art(아스키아트)등에서 큰 서체를 표현할 때에 리용하기도 한다.

> **[주의]**
>
> 다음 문자는 확대출력되지 않는다.
>
> <, >, [,], {, }, |, ~, \, ^, _, ', ", &

banner명령에 별다른 설정값을 주지 않은 경우에는 기본값으로 132렬을 사용하여 확대출력하게 된다. -w설정값을 사용하면 확대하고싶은 크기의 렬수를 지정해줄수 있다. 《Unix》문자렬을 30렬로 확대출력한 실례를 아래에 주었다.

```
kccbbbit@linux:~
# /usr/games/banner -w30 Unix
                  #
          ###############
        ##################
       #####          #
       ##
       ##
       ##
        #
        ##          #
```

```
  ##################
            #
          #
#
#############
#############
         #
         ##
#############
#############
#
#        #
#############  ###
#############   ##
#       #
##      ###
# ##    #####
  ######   #
  #####
#####  ##  #
###     ##
#       #
```

문자를 #으로 표현하기때문에 렬수가 많을수록 원활한 서체를 얻을수 있다.

참　고

아스키아트(ASCII Art)

아스키문자모임을 리용한 도형표현으로서 실시간대화에서 리용하는 표정기호(Smiley)로부터 사진화상변환에 이르기까지 다양하게 표현할수 있다.

basename /bin/basename

파일이름추출

경로와 설정값을 제거한 순수 파일이름만 돌려받는 명령이다.

형식 : basename 경로+파일이름[설정값]

경로+파일이름: 경로를 포함한 파일이름
확장자: 선택설정값으로 확장자까지 제거하려는 경우에만 사용한다. 이 확장자가 파일이름의 맨 마지막에 오는 문자열인 경우, 삭제한다.
—help: 도움말을 보여준다.
—version : 판본정보를 보여준다.

이 명령은 경로를 포함한 파일이름을 인수로 받아 파일의 경로를 제거하고 필요에 따라서는 확장자를 삭제하여 순수 파일이름만 남게 한다.
/etc/issue.net에서 파일경로와 확장자를 지우는 경우를 보자.

```
kcc@linux:~
# basename /etc/issue.net.net
issue
```

[주의]
어떤 파일의 전체 경로를 보려는 경우에는 pwd명령을 사용한다.

스크립트를 작성할 때 어떤 변수가 경로를 포함한 파일이름을 의미한다고 생각해보자. 이 변수에서 파일이름만을 추출하여 사용하고 싶을 때 basename을 사용하면 편리하다.
체계변수중 MAIL은 망접속사용자의 mail파일과 경로를 보관한다.
체계접속(log in)사용자명을 출력할 때에는 다음과 같이 할수 있다.

```
kcc@linux:~
# echo $MAIL
/var/spool/mail/root
# basename $MAIL
root
```

이러한 점을 활용하면 스크립트를 만들 때에 여러가지를 경로에서 추출하여 쉽게 사용할수 있다.

【관련명령문】

dirname : 경로-파일이름에서 경로이름만을 추출한다.

batch

/usr/bin/batch

명령실행의 예약

체계부하가 일정한 값 이하로 되면 예약해둔 명령을 실행한다.

형식 : batch [-V] [-q 대기렬이름] [-f 파일이름] [-mv] [시간]

-q queue : 대기렬(queue)이름을 지정한다. 대기렬이름으로는 a~z, A~Z가 쓰일수 있다. queue이름을 지정하지 않으면 at는 a를, batch는 b를 사용한다. 영문자모순서로 뒤의 이름을 가지는 대기렬은 더 큰 niceness 값을 가진다. 《=》대기렬은 현재 수행되는 작업에 대한 대기렬로 예약되여있다. 대문자이름의 queue에 추가된 작업은 batch의 작업처럼 처리된다.

-m : 출력결과가 없는 경우에도 작업완료때에 사용자에게 우편을 보낸다.

-f 파일명 : 스크립트파일 등을 실행해줄 때에 사용한다.

-v : 작업이 수행될 시간을 보여준다.

체계의 평균부하가 0.8이하로 내려가면 예약해둔 내용을 실행한다.

체계에 부하가 많이 걸리는 명령을 실행할 때에 사용할수 있다. 먼저 현재 체계의 평균부하를 보자.

```
kcc@linux:~

# uptime
3:51am up 19 days, 3:30, 14 users, load average:6.00, 6.00, 5.93
```

[주의]

체계평균부하로는 1분, 4분, 15분간의 평균값을 보여준다.

평균부하가 6정도인것을 확인할수 있다. 체계의 부하를 피해 실행할 스크립트명령을 batch를 리용하여 등록한다.

등록후에 atq명령문을 사용하여 예약된 작업목록을 확인할수 있다.

```
kcc@linux:~
# batch -f system_backup -m now
job 9 at 2004-06-25 04:01
# atq
9  2004-06-25 04:01 b root
```

등록된 시간을 보고 예약한 내용을 확인할수 있다.

【관련명령문】

at : 특정시간에 명령문실행을 예약한다.
atq : 예약된 작업목록을 보여 준다.
atrm : 예약된 작업목록을 삭제한다.
cron : 정기적으로 예약된 작업을 수행한다.
batch : 체계부하가 일정이하일 때에 명령을 수행한다.

참 고

MD5(Message Digest 5)

MD5는 입력자료로부터 128bit의 축소통보문을 만들어 자료무결성을 검증하는데 사용되는 알고리듬으로서 로날드 리베스트교수가 개발하였다. MD5규격은 현재 IETE RFC 1321에 명시되여있다.

이 규격에 따르면 MD5알고리듬에 입력된 어떤 두개의 통보문이 꼭같은 통보문축약을 결과로 내거나 혹은 어떤 통보문축약을 통해 엉뚱한 통보문이 만들어지는것은 《계산적으로 불가능》하다. 용량이 큰 프로그람을 내리적재받았을 때 이 프로그람이 정상인가를 확인하기 위해 이 프로그람의 공식적인 웨브싸이트에서 MD5검사합(MD5 Checksum)을 내리적재받아 비교할수 있다.

bc

/usr/bin/bc

본문계산기

말단장치에서 사용하는 대화형계산기언어.

형식 : bc[-lws][파일...]

-l : 표준수학서고를 정의한다.
-w: POSIX bc에 대한 확장기능에 대하여 경고통보문을 보여준다.
-s : POSIX bc 언어로만 동작하게 한다.

bc는 대화형으로 문장을 실행하는 무한정확도의 수자를 지원하는 언어로서 보통 계산기로 사용한다. 문법은 C언어에서의 연산과 류사하게 사용할수 있으며 명령행설정값을 주면 표준수학서고의 함수를 사용할수도 있다. 그러나 계산기로 사용할 때에는 일반적으로 계산하는 순서대로 식을 입력하여 답을 얻어낼수 있다.

실례로 9870,0000원의 돈을 년리자률 7%로 13년동안 은행에 맡겨놓으면 얼마로 되는가를 계산해보자. 리자를 계산하는 방법에는 여러가지가 있지만 여기에서는 단순히 《원금*((1+리자)^기간)》이라는 식으로 계산해보자.

```
kcc@linux:~

# bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type warranty.
98700000*((1+0.07)^13)
236880000.00
quit
```

우에서 보는바와 같이 목적하는 계산식을 그대로 입력하기만 하면 요구하는 계산결과를 얻을수 있게 한다. 이외에도 bc는 일반계산기보다 훨씬 높은

정확도를 얻을수 있게 한다. bc는 작업을 하다가 다른 계산기를 사용함이 없이 즉시에 사용할수 있도록 할뿐만아니라 복잡하면서도 정교한 계산을 할 때에 도움으로 된다.

참 고

류동소수점과 고정소수점

류동소수점은 일반적인 실수를 가수(mantissa)와 지수(exponent)로 분리한 형태로 나타내는것이다. 이 방식은 한정된 비트수로 정확도가 높은 계산을 하기 위한 것이다. 여기에서 가수는 수의 정밀도를, 지수는 수의 실제크기를 표현한다. 류동소수점수는 1EEE754 규격형식에 의해 옹근수부분과 지수부분으로 나뉘여 기억장소에 보관된다. 이때 전체 62bit중의 첫 비트에는 부호를, 두번째로부터 11번째까지의 비트에는 10의 제곱수를, 나머지에는 유효수자를 보관한다.

반면에 고정소수점은 일반적으로 사용하는 실수와 비슷한데 수를 옹근수부와 소수부로 나누어 사용하는 방법이다. 화상을 처리하거나 렌더링을 하는 도형처리연산에서는 아주 많은 연산을 하게 된다. 앞에서 본바와 같이 류동소수점은 상당히 복잡한 방식이므로 실수형수자를 기억시키는 방식에 있어서 상당히 떠지게 된다. 때문에 빠른 방법으로 실수를 처리하기 위해 등장한 방법이 고정소수점연산이다. 이것은 실수형연산을 앞에서 언급한것과 마찬가지로 옹근수부와 소수부로 나누어 매 옹근수형방식으로 연산할수 있기때문에 빠른 속도로 처리할수 있다.

biff

/usr/bin/biff

우편도착의 알림

우편봉사기를 일정한 간격으로 검사하여 새 우편이 도착하면 통지한다.

형식 : biff [설정값]

y : 우편도착을 통지하여 준다.
n : 우편도착을 통지하지 않는다.

이 명령은 sendmail의 MTA를 사용할 때 유용하게 쓰일수 있는 명령이다.
biff의 설정이 y로 되였으면 새 우편이 도착하였을 때 우편을 보낸 사람과 우편의 첫 부분이 화면에 나타난다. 그러면 현재 biff의 상태를 보도록 하자.

```
kcc@linux:~
# biff
is n
```

대부분의 체계에서는 biff의 표준설정이 off로 되여있다.
이것을 on상태로 만들면 이후에 우편이 올 때마다 통지를 받을수 있다.
on상태로 만들고 확인해보자.

[주의]
현재는 biff를 거의나 사용하지 않는 대신에 xbiff의 GUI응용프로그람이나 우편알림 애플리트를 사용한다.

```
kcc@linux:~
# biff y
# biff
is y
```

y대신 yes를 사용할수도 있다.

bzip2 /usr/bin/bzip2

압축명령문

높은 성능의 압축률을 실현하는 압축편의프로그람이다.

형식 : bzip2[-설정값][파일이름…]

-c -stdout : 파일을 압축하거나 풀어서 표준출력으로 보낸다.

-d -decompress : 압축을 푼다.

-z -compress : 파일을 압축한다.

-t -test : 지정된 파일의 무결성검사만을 하며 압축을 풀지는 않는다.

-f -force : 압축, 해제할 때 같은 이름의 파일이 있는 경우 덧쓰기한다.

-k -keep : 파일을 압축하거나 풀때에 원본파일을 지우지 않는다.

-s -small : bzip2가 작업중에 사용하는 주기억의 량을 줄인다. bzip2작업에 2.5byte만을 사용한다. 이와 같이 작업하게 되면 속도는 2배이상 떠지지만 2.5byte라는 적은 주기억우에서 bzip2를 사용할수 있다.

-q -quiet : 경고통보문이 나오지 않는다.

-v -verbose : bzip2작업사항을 자세히 볼수 있다.

-L -license-V-version : 판본과 사용허가권(License)을 보여준다.

-1~-9 : 압축할 때의 블로크크기를 정해준다.

bzip2는 Burrows-Wheeler블로크정렬검사압축알고리듬과 Huffman코드작성을 사용한것으로서 압축률이 높다. bzip2의 사용법은 GNU gzip에서와 류사하다. 압축된 파일은 압축전의 날자, 권한, 소유자 등의 속성을 그대로 가진다. 그러므로 압축된 파일을 풀었을 때에도 원본파일의 속성을 그대로 유지한다. bzip2에 의하여 압축된 파일의 속성은 .bz2이다.

```
kcc@linux:~
# ls
q-mail.hwp routing
```

우의 두 파일을 bzip2명령으로 압축해보자.

파일압축

```
kcc@linux:~
# bzip2 q-mail.hwp routing
# ls
q-mail.hwp.bz2 routing.bz2
```

파일압축해제

bzip2 -d로 파일을 압축해제한다.

```
kcc@linux:~
#bzip2 -d q-mail.hwp.bz2
# ls
q-mail.hwp routing.bz2
```

bzip -d는 bunzip2명령과 같은 명령이다.

```
kcc@linux:~
# bunzip2 routing.bz2
# ls
q-mail.hwp routing
```

【관련명령문】

compress : 압축편의프로그람. 확장자는 . Z

gzip : 표준 GNU/UNIX 압축편의프로그람. 확장자는 .gz

참 고

Huffman 코드작성

자료코드를 통계적으로 검사하여 압축하는 방식이다. 빈도가 높은 코드는 적은 수의 비트로 표현하고 빈도가 낮은 코드는 상대적으로 많은 비트로 표현하여 압축한다.

biz2 압축파일이 파괴된 경우

bzip2recover 를 사용하여 파괴된 압축파일을 복구한다.

bzip2recover [파괴된 압축파일이름]

```
kcc@linux:~
# bzip2recover test.bz2
bzip2recover 1.0: extracts blocks from damaged
.bz2 files.
bzip2recover:searching for block baundaries...
  block 1 runs from 80 to 607
bzlp2recover: splitting into blocks
  writing block 1 to rec0001test.bz2' ...
bzip2recover: finished
```

복구된 파일이 rec0001test.bz2 로 만들어진것을 확인할수 있다.

rec0001test.bz2 의 무결성을 -tv(검사 및 작업내용보기)에 의해 검사한다.

```
kcc@linux:~
# bzip2 -tv rec0001test.bz2
rec0001test.bz2: ok
```

파괴된 압축파일이 정상적으로 복구되였음을 확인할수 있다.

C

cal /usr/bin/cal

력서보기

체계에서 력서의 기능을 수행하는 명령문

형식 : cal[설정값][[month]year]

-j : 양력으로 출력.
-y : 현재 년도의 모든 달을 보여준다.
-V : 판본정보를 보여준다.

사용할수 있는 년도는 양력 원년으로부터 9999년까지이다. 9999년이후의 력서를 나타낼수 없는것은 유감이지만 이 문제는 9999년이 오기전에 수정대책될것이다. 1999년을 보려고 할 때에는 《cal 99》가 아니라 《cal 1999》라고 입력해야 한다. 아무런 명령행인수도 없으면 체계의 현재 달을 보여준다. 2004년 6월을 실례로 보자.

```
kcc@linux:~

# cal
        2004년 6월
일  월  화  수  목  금  토
            1   2   3   4   5
6   7   8   9   10  11  12
13  14  15  16  17  18  19
20  21  22  23  24  25  26
27  28  29  30
```

[주의]

영문력서를 보려는 경우에는 LANG=en을 실행한 후에 력서를 실행한다. 력서의 언어환경은 사용자자신의 환경변수에 따른다.

특정한 년도와 월을 지정하여 찾으려는 력서를 볼수도 있다.

실례로 1973년 8월을 보기로 하자.

```
kcc@linux:~
# cal 8 1973
        1973년 8월
일  월  화  수  목  금  토
            1       2   3   4
5   6   7   8   9   10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30  31
```

월을 표시하지 않고 2003년 력서를 보려고 하는 경우에는 다음과 같이 할 수 있다.

```
kcc@linux:~
# cal 2003
                          2003

          1월                                 2월
일  월  화  수  목  금  토          일  월  화  수  목  금  토
            1   2   3   4                                   1
5   6   7   8   9   10  11          2   3   4   5   6   7   8
12  13  14  15  16  17  18          9   10  11  12  13  14  15
19  20  21  22  23  24  25          16  17  18  19  20  21  22
26  27  28  29  30  31              23  24  25  26  27  28

          3월                                 4월
일  월  화  수  목  금  토          일  월  화  수  목  금  토
                        1                   1   2   3   4   5
2   3   4   5   6   7   8           6   7   8   9   10  11  12
9   10  11  12  13  14  15          13  14  15  16  17  18  19
16  17  18  19  20  21  22          20  21  22  23  24  25  26
23  24  25  26  27  28              27  28  29  30  31
29  30  31
```

5월

일	월	화	수	목	금	토
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

6월

일	월	화	수	목	금	토
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

7월

일	월	화	수	목	금	토
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

8월

일	월	화	수	목	금	토
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

9월

일	월	화	수	목	금	토
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

10월

일	월	화	수	목	금	토
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

11월

일	월	화	수	목	금	토
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

12월

일	월	화	수	목	금	토
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

이 내용을 한번에 인쇄기로 출력하여 1년 력서로 만들려면 다음과 같이 입력하면 된다.

```
# cal 2003 l lpr
```

【관련명령문】

date : 오늘의 날자 혹은 지정한 날의 날자를 주어진 형식대로 출력한다.

cardctl

/sbin/cardctl

PCMCIA기판의 관리

학습장형콤퓨터는 물론 일반체계에서도 PCMCIA기판설정을 관리하는 명령이다.

형식 : cardctl[설정값]명령문

status[소케트] : 현재 소케트상태정보를 보여준다.

config[소케트] : 현재 소케트설정을 보여준다.

ident[소케트] : 카드설정정보를 보여준다. 제품정의, 제작회사, ID 코드등을 포함한다.

suspend [소케트] : 체계끝내기(Shutdown)를 하고 전원을 끈다.

resume [소케트J : 소케트에 전원을 넣고 사용을 위해 재설정을 한다.

reset [소케트] : 소케트에 재설정신호를 보낸다.

eject [소케트] : 체계제거준비후에 카드를 제거한다.

insert [소케트] : 체계삽입준비후에 카드를 삽입한다.

scheme[소케트] : scheme 이름이 지정되지 않으면 현재 PCMCA 설정 scheme 를 보여준다. scheme 이름이 지정되면 모든 PCMCA 장치를 새로운 scheme 에 맞게 재설정한다.

-v : 판본정보를 보여준다.

-c config : /etc/pcmcia 대신 지정한 등록부에서 설정자료기지와 설정스크립트를 불러온다.

-f scheme : /var/lib/pomcia/scheme 대신 지정한 파일에 현재 설정 scheme 를 보관한다.

아래의것은 cardctl명령에 사용하는 설정등록부이다.

```
kcc@linux:~

# ls -F /etc/pcmcia/
cis/         ftl.opts   memory*       parport.opts   serial.opts
conflg*      ide*       memory.opts   scsi*          shared
config.opts  ide.opts   network*      scsi.opts      wireless*
ftl*         isdn*      parport*      serlal*        wireless.opts
```

```
kcc@linux:~

# cat /etc/pcmcia/config

device "3c589_cs"
  class "network" module "3c589_cs"

device "avm_cs"
  class "isdn" module "avm_cs"

device "avmal_cs"
  class "isdn" module "net/slhc", "misc/isdn", "misc/hisax"
 opts "type=26 protocol=2", "avma1_cs"
```

config파일에서는 설정할수 있는 PCMCIA기판목록들을 확인해볼수 있다. cardetr에 status명령을 사용하면 PCMCIA기판의 상태를 확인할수 있다. 만일 인식한 PCMCIA기판이 없는 경우에는 다음과 같은 통보문이 나온다.

```
kcc@linux:~

# cardctl status
no pcmcia driver in /proc/devices
```

장치가 제대로 설치된 경우에는 아래와 같이 현재 상태와 정보를 보여준다.

```
kcc@linux:~

# cardctl status
Socket 0:
5V 16-bit card present
Function 0: ready, write protect
Socket 1:
no card
```

그러면 config명령으로 현재의 설정상태를 확인해보자.

```
kcc@linux:~
# cardctl config
Socket 0:
Vcc=5.0 Vpp1=5.0, Vpp2=5.0
Socket 1:
not configured
```

ident명령은 기판설정정보와 함께 제품정의, 제작회사의 고유코드 등을 보여준다.

```
kcc@linux:~
# cardctl ident
Socket 0:
product info : "RP" "1625B Ethernet NE2000
Compatible", "EP401"," "
function : 6(network)
Socket 1:
no product info available
```

참 고

PCMCIA

PCMCIA란 Personal Computer Memory Card International Association의 략자로서 국제콤퓨터기억기기판협회를 말하는데 개인용콤퓨터에 사용되는 각종 기판제품의 표준규격취급에 관하여 1989년에 구성된 국제적인 규격제정협회이다.

PCMCIA기판이란 PCMCIA에서 추천한 기판제품을 의미한다. PCMCIA기판은 보통 신용기판정도의 크기로서 현재 약 200종류이상이 판매사용되고있다.

PCMCIA기판은 간단한 자료보관장치로부터 정교한 기술을 요구하는 분야에 이르기까지 다양하게 쓰인다. 실례로 콤퓨터와 통신장치와의 련결에 널리 사용될뿐만 아니라 GPS체계 등에도 사용되며 최근에는 이동체기구(Mobile Device)에도 사용되고있다.

cardmgr

/sbin/cardmgr

PCMCIA장치관리자

PCMCIA기판에 대한 설정과 모듈관리를 담당하는 명령이다.

형식 : cardmgr[설정값]

-v : 판본정보를 보여준다.
-q : 카드를 삽입할 때에 경고음을 내지 않는다.
-v : 정상동작을 하는동안 보다 많은 정보를 보여준다.
-d : 구동기모듈을 적재할 때에 관련모듈 의존성이 따라온다.
(cardmgr는 insmod지령대신에 modprobe지령을 사용한다).
-f : foreground방식으로 실행된다.
-o : 카드설정 후에 탈퇴한다(one pass).
-c configpath : /dtc/pcmcia대신에 지정한 등록부에서 설정자료기지와 설정스크립트를 불러온다.
-m modpath : /lib/modules/핵심부모듈등록부대신에 지정한 등록부에서 핵심부모듈을 적재한다.
-p pidfile : /var/run/cardmgr.pid대신 지정한 파일에 cardmgr프로쎄스 PID를 보관한다.
-s stabfile : /var/lib/pcmcia/stab대신 지정한 파일에 현재 소케트정보를 보관한다.

cardmgr명령은 /etc/rc.d/init.d/pcmcia 데몬과 관련이 있다.

/sbin/cardmgr는 소케트의 기판삽입과 제거를 감시하여 삽입이 이루어지는 경우에는 /etc/pcmcia/config안의 기판정의목록을 확인해보고 적절한 모듈을 올려준다. 대부분이 데몬만 실행하여주면 스크립트가 실행되여 PCMCIA기판을 자동인식하게 되여있다.

cardctl 명령에서 설명한 cardctl ident명령으로 PCMCIA인식여부를 확인할수 있다.

```
kcc@linux:~

# cardctl ident
Socket 0:
no product info available
Socket 1:
product info: "RP", "1625B Ethernet NE2000 Compatible", "EP401", " "
function: 6(network)
```

현재 소케트 0번에는 아무것도 없고 1번에는 NE2000 Compatible 국부망기판이 접속되여있다.

만일 제대로 인식하지 못하면 /etc/pcmcia/config파일을 확인하여보고 해당한 체계에 맞게 편집해야 한다.

```
kcc@linux:~

card "1625B Ethernet NE2000 Compatible"
version "RP", "1625B Ethernet NE2000 Compatible", "EP401"
bind "pcnet_cs"
```

설정이 끝났으면 pcmcia데몬을 기동시킨다.

```
#/etc/rc.d/init.d/pcmcia restant
```

PCMCIA와 관련있는 파일은 다음과 같다.

/etc/pcmcia/config	기판설정자료기지
/etc/pcmcia/config.opts	PCMCIA장치를 위한 local source setting
/var/run/cardmgr.pid	현재 cardmgr프로쎄스PID
/var/lig/pcmcia/stab	매 소케트에 대한 현재 기판과 장치정보

cat

/bin/cat

파일보기

본문파일내용을 표준출력으로 보여준다.

형식 : cat[설정값][파일]

-A, —show-all : -vET와 같다.
-b, —number-nonblank : 공백행을 제외한 파일내용에서 행의 맨앞에 1부터 시작하는 행수를 보여준다.
-e : -vE와 같다.
-E, —show-ends : 각 행의 끝에 $로 끝행을 나타낸다.
-n, —number: 모든 행의 앞에 1부터 시작하는 행수를 보여준다.
-s, —squeeze-blank : 공백행의 행수를 하나이상 보여주지 않는다.
-t : -vt와 같다.
-T, —show-tabs : tab문자를 ^I로 보여준다.
-v, —show-nonprinting : 개행문자와 tab문자를 제외한 ^I와 M표시를 보여준다.
-help : 도움말을 보여준다.

cat는 파일의 내용을 출력하여 보여주는 대표적인 명령문이다.

cat는 파일내용을 한번 보여주기만 할뿐이고 많은 행의 내용은 제대로 볼 수 없게 한다.

이때에는 보통 more명령, less명령, tail명령을 사용하든가 출력내용을 관기호(pipe(|))를 통해 다른 프로그람과 조합하여 사용하여야 한다.

다음의 query파일을 보자. 한행이 길기때문에 한행인지 잘 구별되지 않는다.

[주의]

관기호는 앞명령의 표준출력을 뒤명령의 표준입력으로 넘겨주는 역할을 한다.

```
kcc@linux:~

# cat query
select date_format(uregdate, '%Y%m') date,count(uid) from cst_users u,
cst_productrgs p, cst_productregdetail d
where u.uno=p.prefuno
and p.pregno=d.preggno
group by date
```

-n설정값은 행앞에 행번호를 붙여주는 역할을 한다. 이 파일은 총 4행임을 알수 있다.

```
kcc@linux:~

# cat-n query
  1  select data_format(uregdate, '%Y%m') date, count(uid) from cst_users
     u.cat_productregs p,cst_productregdetail d
  2 where u.uno=p.preguno
  3 and p.pregno=d.preggno
  4 group by date
```

[주의]
단순히 몇행으로 되였는가를 알기 위해서는 wc -l명령을 사용한다.

행수가 많은 경우에 파이프로 more명령을 실행하여 한 페지씩 볼수 있다. 다음 페지를 보려면 공백건(Space Key)을 입력하며 한행씩 내려가려면 실행건(Enter Key)을 입력한다. 물론 more만 실행하여도 같은 결과를 볼수는 있다.

```
kcc@linux:~

# cat /etc/httpd/conf/httpd.conf | more
#
# Based upon the NCSA server configuration files originally by Rob cCool.
#
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See < URL: http://www.apache.org/docs/> for deailed information about
```

```
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# After this file is processed, the server will look for and process
# /etc/httpd/conf/srm.conf and then/etc/httpd/conf/access.conf
# unless you have overridden these with ResourceConfig and/or
# AccessConfig directives here.
#
# The configuration directives are grouped into three basic sections:
# 1. Directives that control the operation of the Apache server process as a
#    whole (the 'global environment').
# 2. Directives that define the parameters of the 'main' or 'default' server,
#    which responds to requests that aren' t handled by a virtual host.
—More—
```

마지막으로 cat는 간단한 행편집기의 역할을 할수 있다. 행편집기(Line Editor)란 현재 입력하고있는 행은 수정할수 있지만 이미 지나온 행은 수정할수 없는것으로서 지금은 거의나 쓰이지 않는다. 사용방법은 다음과 같다.

```
# cat>[파일이름]
```

그러면 실지로 파일을 만들어 보관해보자. 파일을 다 쓰고나서는 Ctrl+D를 눌러 cat명령을 완료한다.

```
kcc@linux:~
# cat > nae.txt
You Still my No.1,
이국의 들가에 피여 난 꽃도
내 고향 꽃보다 곱지 못했소
돌아보면 세상은 넓고 넓어도
내 사는 내 나라 제일로 좋아
Ctrl+D
```

노래가사를 입력하여 보았다. 그러면 제대로 보관되였는가를 새로 만들어진 nae.txt 파일을 읽어 확인해보기로 하자.

```
kcc@linux:~

# cat nae.txt
You Still my No.1,
이국의 들가에 피여 난 꽃도
내 고향 꽃보다 곱지 못했소
돌아보면 세상은 넓고 넓어도
내 사는 내 나라 제일로 좋아
```

제대로 보관된것을 알수 있다. 그러나 이 경우에 한가지 주의할 점이 있다. 다음의 내용을 보자. 다음과 같이 사용하면 100m.txt 파일의 내용을 화면에 출력하는 대신에 nae.txt파일에 입력하여 보관한다.

```
kcc@linux:~

# cat 100m.txt > nae.txt
# cat nae.txt,
이국의 들가에 피여 난 꽃도
내 고향 꽃보다 곱지 못했소
```

앞에서 작성한 nae.txt의 내용이 사라지고 100m.txt의 내용이 보관되여 있다. 그러면 먼저 보관된 내용의 뒤에 새로운 내용을 보관하려면 어떻게 하여야 하겠는가? 이때에는 > 를 두개 사용하여 문제를 해결할수 있다.

```
# cat sarah.txt >>nae.txt
```

【관련명령문】

more : 한 페지씩 파일을 볼수 있게 한다.

tail : 파일의 끝부분부터 지정한 행수만큼 보여준다.

less : more 와 같은 기능을 하지만 추가적인 기능들이 있다.

cd

내부명령문

파일등록부이동

작업하려는 등록부에로 이동한다.

형식 : cd[등록부이름]

cd는 등록부를 이동하는 명령문이다. 단순하지만 가장 기본적인 명령으로서 자주 사용된다.

명　　령	설　　명
cd 등록부	이동하려는 등록부에로 이동한다.
cd .	현재 등록부
cd ..	웃준위등록부로 이동한다.
cd $	변수명변수에 지정된 등록부에로 이동한다.
cd /	가장 웃준위의 등록부에로 이동한다.
cd~/cd $HOME/cd	사용자의 home등록부에로 이동한다.
cd~사용자명	지정된 사용자의 home등록부에로 이동한다.

pwd로 현재 파일등록부를 확인한다. 한 단계우의 웃준위등록부로 이동하거나 어느 위치에서도 사용자(User Account)의 HOME등록부에로 이동할 수 있다.

```
kcc@linux:~
# pwd
/usr/local
# cd ..
# pwd
/usr
#cd ~bulletin
# pwd
/home/magazine
```

cfdisk

/sin/cfdisk

구획설정

유표안내방식의 구획설정편의프로그람

형식 : cfdisk[설정값][장치]

-a : 현재 구획을 유표(Cursor)로 선택한다.
-c cylinders : 실린더개수를 지정한다.
-h heads : 머리부의 개수를 지정한다.
-s sector-per-track : 기록자리길당 분구(Sector/Track)수를 지정한다.
-z : 구획정보를 읽지 않고 시작한다.
-P format : r(raw data), s(sector order), t(raw format)의 형식에 따라 구획표를 보여준다.

cfdisk는 유표안내방식의 구획설정편의프로그람이다. Linux에서는 기본구획설정명령인 fdisk이외에 cfdisk나 설치할 때에 볼수 있는 Disk Druid와 같은 도형방식의 프로그람을 지원한다.

[주의]
자료가 있던 디스크령역에 새로 구획을 설정한 다음 다시 재기동하면 이미 있던 구획표(Partition Table)가 삭제되며 자료를 사용할수 없게 된다.

cfdisk명령으로 구획(Partition)을 설정하려면 다음과 같이 실행한다.

```
kcc@linux:~
# cfdisk /dev/hda
cfdisk 2.11n
                              Disk Drive:/dev/hda
                              Size : 8455200768 bytes
 Heads : 255  Sectors per Track : 63  Cylinders: 1027
Name  Flags  Part Type  FS  Type    [Label]     Size(MB)
```

```
hda1                 Primary  Linux swap           271.44
hda2        Boot      Primary  Linux ext2    [/]      8175.93
[Bootable] [ Delete ] [ Help ] [Maximize] [Print]
            [ Quit ] [ Type ] [ Units ] [ Write ]
             Toggle bootable of the current partition
```

실행후에는 다음과 같은 명령들을 사용할수 있다.

?	도움말
b	선택된 구획으로 기동할수 있도록 설정
d	선택된 구획의 삭제
g	전문가방식
h	도움말
n	새로운 구획의 생성
p	현재의 구획정보를 출력
q	완료
t	파일체계의 완료변경
W	보관

【관련명령문】

fdisk: 구획을 설정하기 위한 명령문이다.

참 고

MBR(Master Boot Reoord)

하드디스크가 어떻게 나누어져 있는가에 대한 정보를 담고있는 하드디스크의 첫번째 분구(Sector)이다.

MBR는 콤퓨터가 처음 기동될 때에 BIOS가 읽어들여 시작하는 분구로서 구획정보를 읽어들이고 어떤 구획이 기동가능한 구획인가를 확인하여 각 구획의 기동분구(boot sector)를 읽어들이는 프로그람을 포함하고있다.

chage

/usr/bin/chage

암호유효기일의 지정

체계보안의 목적으로 사용자암호의 유효기일을 설정 및 변경하는 명령문이다.

형식 : chage[설정값]user

-m [최소날자] : 암호변경후에 다시 변경할수 있는 최소날자를 지정한다. 0값을 주면 매번 암호를 변경하여야 한다.

-M [최대날자] : 암호가 유효한 최대날자를 지정한다.

-d [마지막날자] : 암호의 마지막 변경날자를 YYYY-MM-DD형태로 나타낸다.

-E [유효기일] : 사용자식별이름이 더 이상 접근불가능한 날자를 지정한다. 이후에 암호는 잠금상태로 된다.

-I [잠금날자] : 식별이름의 암호유효기일이 다 되여 식별이름을 사용할수 없게 되는 날자를 지정한다. 그 이후에 암호를 잠금상태로 만든다.

-w [경고날자] : 암호가 유효기일이 지나기전에 안내통보문은 날자를 지정한다.

-l user : 사용자의 암호유효기일정보를 보여준다.

chage는 사용자의 암호유효기일정보를 보거나 유효기일을 설정변경할수 있게 한다. 이것은 /etc/passwd와 /etc/shadow파일을 참고한다.

체계보안의 측면에서 볼 때 암호관리는 매우 중요한데 많은 사용자들에 대한 관리는 품이 많이 들고 복잡한 작업으로 될수 있다.

chage명령은 미리 지정한 날자에 의해 그 날자가 지나면 암호를 변경하여 보다 효률적인 체계관리를 할수 있도록 도움을 준다.

먼저 admin식별이름에 대한 유효기일과 기타 정보를 보기로 하자.

```
kcc@linux:~
# chage -l admin
Minimum:       0
Maximum:    99999
Warning:      7
Inactive:    -1
Last change:     2월 19, 2004
Password Inactive: Never
Account Expires: Never
```

그러면 admin의 식별이름암호유효기일을 최대날자수: 7일, 최소날자수: 1일, 경고번호: 7로 지정해보자.

```
kcc@linux:~
# chage-M 7 -m 1-w7 admin
# chage -1 admin
Minimum:      1
Maximum:        7
Warning:      7
Inactive:     -1
Last Change:             5월 11, 2004
Password Expires:        5월 18, 2004
Password Inactive:       Never
```

-I설정값에 0을 주면 설정한 유효기일이 지난 다음에는 식별이름을 사용할 수 없게 한다.

```
# chage -I 0 admin
```

-E설정값을 사용하여 admin 식별이름을 2004년 6월 30일이후에 잠그도록 하자.

[주의]

한번 잠긴 파일은 관리자가 잠금해제를 시켜주지 않는 한 사용할수 없다.

```
# chage -E 2004-06-30 admin
```

admin식별이름을 잠금해제시켜본다.

```
# chage -E 0 admin
```

chattr

/usr/bin/chattr

속성변경

파일체계의 파일속성을 변경하여 관리자가 실수로 인해 파일을 파괴시키지 않도록 보호한다.

형식 : chattr[설정값]속성파일...

-R : 등록부와 그 이하의 모든 속성을 변환시킨다.
-V : 변화된 속성에 대하여 자세히 출력한다.
-v : 판본정보를 출력한다.
a : 파일을 추가방식으로만 열수 있다.
c : 핵심부에 의해 디스크상에 압축상태로 보관된다.
d : dump 지령수행때에 여벌복사되지 않는다.
I : 파일을 수정할수 없다. 오직 super user만이 다시 이 속성을 변경시킬수 있다.
s : 파일이 지워질 때에 일단 블로크들이 모두 0으로 된 다음 디스크에 기록된다.
S : 파일이 수정될 때 그 변화가 디스크상에 동기화된다.

chattr는 파일체계에서의 파일속성을 변경하는 명령문이다. 이것은 반드시 보호되여야 하는 파일들을 실수로 지우거나 덧쓰기하는 경우가 없도록 하도록 하기 위해 필요하다.

+속성은 파일에 속성을 추가하며 -속성은 파일에서 속성을 제거한다.

=속성은 파일이 오직 주어진 속성만 가지도록 한다.

[주의]

이 명령문은 reiser체계에서는 제대로 동작하지 않는 경우가 있다. 그러나 일반적으로 ext2, ext3파일체계를 사용하므로 대부분은 실행된다.

chattr +i설정으로 /etc/passwd에 i속성을 부여해보자.

이것은 파일삭제, 이름변경, 내용추가, 련결생성 등을 모두 불가능하게 만든다.

```
#chattr+i/etc/passwd
```

lsattr명령으로 변경된 속성들을 확인할수 있다.

```
kcc@linux:~
# lsattr /etc/passwd
----i--------/etc/passwd
```

+i속성이 부여된 /etc/passwd파일을 삭제하여보자. 속성을 제거하기전에는 파일삭제를 할수 없으며 파일내용의 변경도 불가능하다.

vi편집기로 파일을 열어도 읽기전용으로서 수정이 불가능하게 된다.

[주의]

경험이 적은 관리자는 외부공격자가 침입하여 설치하고 나간 불법파일에 파일삭제 및 파일내용변경을 불가능하게 하는 속성이 걸려 있는 경우에 그를 제거하지 못하여 곤난해할 때가 있다. 수상한 파일을 관리자의 권한으로도 삭제하거나 수정할수 없는 경우에는 반드시 lsattr을 사용하여 확인해보아야 한다.

```
kcc@linux:~
# rm -rf /etc/passwd
rm: cannot unlink /etc/passwd : Operation
not permitted
```

그러면 이 -i 속성을 /etc/passwd 파일에서 제거하여보자.

```
kcc@linux:~
# chattr -i /etc/passwd
# lsattr /etc/passwd
-------------/etc/passwd
```

우와 같이 한 이후에는 이 파일을 수정하거나 제거할수 있다.

【관련명령문】

lsattr : Linux 파일체계의 파일속성을 보는 명령문

chfn /usr/bin/chfn

사용자정보의 변경

일일히 암호파일을 변경하지 않고도 finger명령문을 사용하였을 때에 볼수 있는 사용자 기본정보를 변경하는 명령이다.

형식 : chfn[설정값]사용자이름

user name: 정보를 변경하려는 사용자이름. 공백인 경우에 자신의 식별이름이 변경된다.

-f, —full-name: 사용자의 전체 이름을 변경한다(식별이름과 다름).

-h, —home-phone : 사용자의 집전화번호를 변경한다.

-o, —office : 사용자의 직장명을 변경한다.

-p, —office-phone : 사용자의 직장전화번호를 변경한다.

-u, —help : 도움말을 보여준다.

-v, —version : 판본정보를 보여준다.

[주의]

/etc/passwd는 일반사용자가 직접 수정할수 없다.

이 명령문은 등록되여있는 사용자의 정보를 변경할 때에 사용한다.

봉사기관리자가 식별이름에 등록된 사용자의 정보를 변경할 때에 사용할수 있다.

chfn명령은 /etc/passwd에 보관된 정보를 변경하므로 본문편집기로 이 파일을 열어 변경해주어도 된다.

finger명령으로 사용자정보를 확인할수 있다.

관리자가 사용자의 정보를 변경하려는 경우에 다음과 같이 한다.

여기에서는 《first》라는 사용자의 정보를 바꾸어보도록 한다.

```
kcc@linux:~

#chfn admin
changing finger information for first
Name[1번수] : 최철호              →이름의 변경
Office[KCC] : KCCSamilpo          →직장의 변경
Office Phone[] : 721-2826         →직장전화번호의 변경
Home Phone[]: 521-4584            →집전화번호의 변경
```

사용자정보중에서 이름만 《1번수》로 변경하려는 경우에는 다음과 같이 할수 있다.

```
kcc@linux:~

# chfn -h 《1번수》 first
changing finger information for first
Finger information changed.
```

변경된 정보를 확인하려면 finger명령을 리용한다.

```
kcc@linux:~

#finger first : 사용자정보확인
Login : first                    Name : 1 번수
Directory:/home/first            Shell : /bin/bash
Office: KCCSamilpo, 721-2826     Home Phone : 521-4584
```

사용자이름, 직장전화번호, 집전화번호 등이 변경된것을 확인할수 있다.

【관련명령문】

finger : 사용자정보의 확인
chgrp : 파일의 집단변경
chmod : 파일의 권한변경
chown : 파일의 소유자변경
chsh : 쉘변경

chgrp

/bin/chgrp

집단변경

파일의 소유집단을 변경하는 명령문이다. 주로 chmod와 함께 사용하여 파일을 사용하는 접근권한을 설정할 때에 사용한다.

형식 : chgrp[설정값]집단명 파일이름

group name : 새로운 집단명

filename : 집단을 변경하려는 파일이름

-c, -changes : 집단이 변경되는 파일만 보여준다.

-f, -silent, -quiet : 집단이 변경되지 않는 경우에도 오유를 보여주지 않는다.

-help : 도움말을 보여준다.

-R, -recursive : 하위등록부안의 모든 등록부/파일의 집단을 변경한다.

-reference=filename : 지정한 파일에서의 집단을 변경한다.

-v, -verbose : 집단변경을 상세히 보여준다.

-version : 판본정보를 보여준다.

이 명령문은 파일을 새로운 집단으로 변경한다. 새로운 집단의 고유번호나 집단이름으로 변경할수 있으며 /etc/group에서 새로운 집단에 대한 정보를 확인할수 있다. 집단의 변경은 파일의 소유자나 관리자만이 할수 있다.

first등록부와 그 아래준위등록부를 모두 paduk집단으로 바꾸고 변경된 집단을 확인해보자.

```
kcc@linux:~

# chgrp -R paduk first/
# ls -al
drwxr-xr-x   5   root   root   4096    1월 26일 10:52 .
drwxr-xr-x   5   root   root   4096    1월 26일 10:53 ..
drwxr-xr-x   5   root   fish   4096    3월 1일  10:53 first
```

참 고

기본집단

/etc/group파일을 화면출력하여 확인해보면 사용자가 지정해주지 않은 집단이 눈에 많이 띄운다. 아래에 보여준것들은 설치중에 자동적으로 설정되는 기본집단이다.

집 단	GID	포함요소
root	0	root
bin	1	root, bin, daemon
daemon	2	root, bin, daemon
sys	3	root, bin, adm
adm	4	root, adm, daemon
tty	5	
disk	6	root
lp	7	daemon, lp
mem	8	
kmem	9	
wheel	10	root
mail	12	mail
news	13	news
uucp	14	uucp
man	15	
games	20	
gopger	30	
dip	40	
ftp	50	ftp
nobody	99	
users	100	

우와 같은 기본집단은 대체로 체계의 기본사용자와 관련있는것으로서 특정한 프로그람을 구동하기 위해 사용되는 경우가 대부분이다. 이 내용은 어떤 프로그람을 설치하는가에 따라 달라지지만 대부분의 경우에 root와 deamon, sys, adm, disk, bin, whee 등의 root와 관련있는 집단은 어떻게 설치하든간에 자동적으로 설정되게 된다. 이와 같은 기본집단은 일반사용자에게 권한으로 주어서는 안된다.

chkconfig

/sbin/chkconfig

구동프로그람설정

체계를 기동할 때에 실행준위별로 자동실행할 봉사를 확인해보고 갱신하는 명령이다.

형식 : chkconfig --list[이름]
chkconfig --add 이름
chkconfig --del 이름
chkconfig[--level 준위]이름<on|off|reset>
chkconfig[--level 준위]이름

기동할 때에 봉사할 데몬은 /etc/rc.d 매 등록부에 모여져 있다. rc뒤에 붙은 수자는 t실행준위번호, rc0.d는 실행준위 0이며 rc1.d는 1, rc5.d는 실행준위 5에 대한 등록부이다.

먼저 rc5.d 등록부를 보자. 여기에서 K로 시작하는 파일은 체계를 시작할 때에 시작하지 않는 데몬이며 s로 시작하는 파일이 기동때에 봉사할 데몬이다. s다음의 수자는 시작할 데몬의 순서이다.

[주의]
ntsysv 명령을 사용하면 안내방식으로 편리하게 설정할수 있다.

```
kcc@linux:~

# ls /etc/rc.d/rc5.d/
K00linuxconf   K20dhcpd     K45arpwatch  K74apmd     K96pcmcia
K00webmin      K20iscsi     K46radrd     K74nscd     S08ipchains
K01adsl        K20nfs       K50snmpd     K74utpd     S10network
K01kdcrotate   K20rstatd    K50tux       K74ups      S12syslog
K05anacron     K20rusersd   K50xinetd    K74ypserv   S17keylable
K05atd         K20rwalld    K54pxe       K74ypxfrd   S28autofs
K08vmware      K20rwhod     K55routed    K75gated    S55named
K10medusa      K25squid     K60crond     K75netfs    S55sshd
K12FreeWnn     K28amd       K61ldap      K80random   S60lpd
```

```
K12canna         K30mcserv       K61ptal-init   K86nfslock   S78mysqld
K12kWnn          K30sendmail     K65identd      K87portmap   S85httpd
K15gpm           K34yppasswdd    K65kadmin      K89bcm5820   S90xfs
K15proftpd       K35smb          K65kprop       K92iptables  S99local
K16rarpd         K35vncserver    K65krb524      K95kudzu
K20bootparamd    K40mars-new     K65krb65kdc    K95reconfig
K20dhcp-rdlay    K44rawdevices   K73ypbind      K96irda
```

기동할 때에 시작할 데몬파일이름을 K로 시작하는 파일에서 S로 시작하는 파일이름으로 변경하여 설정할수도 있지만 chkconfig명령을 사용하면 손쉽게 변경할수 있으며 많은 기능을 볼수 있다.

```
kcc@linux:~

# chkconfig --list | more
keytable    0:해제  1:활성  2:활성  3:활성  4:활성  5:해제  6:해제
atd         0:해제  1:해제  2:해제  3:해제  4:활성  5:해제  6:해제
kdcrotate   0:해제  1:해제  2:해제  3:해제  4:해제  5:해제  6:해제
syslog      0:해제  1:해제  2:활성  3:해제  4:활성  5:활성  6:해제
gpm         0:해제  1:해제  2:활성  3:해제  4:활성  5:해제  6:해제
kudzu       0:해제  1:해제  2:해제  3:활성  4:활성  5:해제  6:해제
vmware      0:해제  1:해제  2:해제  3:해제  4:해제  5:해제  6:해제
sendmail    0:해제  1:해제  2:활성  3:해제  4:활성  5:해제  6:해제
netfs       0:해제  1:해제  2:해제  3;해제  4:활성  5:해제  6:해제
network     0:해제  1;해제  2:활성  3:활성  4:활성  5:활성  6:해제
random      0:해제  1:해제  2:활성  3:해제  4:활성  5:해제  6:해제
rawdevices  0:해제  1:해제  2:해제  3:활성  4:활성  5:해제  6:해제
pcmcia      0:해제  1:해제  2:활성  3:활성  4:활성  5:해제  6:해제
apmd        0:해제  1:해제  2:활성  3:해제  4;활성  5:해제  6:해제
ipchains    0:해제  1:해제  2:활성  3:활성  4:활성  5:활성  6:해제
iptables    0:해제  1:해제  2:활성  3:해제  4:활성  5:해제  6:해제
crond       0:해제  1:해제  2:활성  3:해제  4:활성  5:해제  6:해제
anacron     0:해제  1:해제  2:해제  3:해제  4:해제  5:해제  6:해제
lpd         0:해제  1:해제  2:활성  3:활성  4:활성  5:활성  6:해제
xfs         0:해제  1:해제  2:활성  3:활성  4:활성  5:활성  6:해제
—More--
```

우의 목록중에서 keytable데몬이 실행준위 5에서 해제되여있다. 이것을 활성화하도록 해본다.

```
# chkeonfig --level 5 keytable on
```

—list설정은 봉사정보를 볼수 있게 한다.

```
kcc@linux:~
# chkconfig —list keytable
keytable 0:해제 1:활성 2:활성 3:활성 4:활성 5:활성 6:해제
```

설정한 실행준위 5에서의 데몬봉사를 다시 해제하여보자.

```
kcc@linux:~
# chkconf1g --level 5 keytable off
# chkconfig --list keytable
keytable  0:해제 1:활성 2:활성 3:활성 4:활성 5:해제 6:해제
```

—del명령에 의해 지정한 데몬봉사와 관련된 모든 실행준위를 해제할수도 있다.

```
kcc@linux:~
# chkconfig —del keytable
# chkconfig —list keytable
keytable 0:해제 1:해제 2:해제 3:해제 4:해제 5:해제 6:해제
```

그러면 다시 —add명령으로 1,6준위를 제외한 모든 준위를 활성화시키자.

```
kcc@linux:~
# chkconfig —add keytable
# chkconfig —list keytable
keytable  0:해제 1:활성 2:활성 3:활성 4:활성 5:활성 6:해제
```

【관련명령문】

ntsysv : 기동시에 봉사할 안내형관리편의프로그람

chmod

/bin/chmod

권한변경

파일의 권한을 변경하는 명령문이다.

형식 : chmod[설정값]방식 파일이름

mode : 새로운 권한(permissions)
filename : 권한을 변경하려는 파일이름
-c, -changes : 변경된 파일들에 대한 정보를 보여준다.
-f, -silent, -quiet : chomod로 변경되지 않은 파일에 대한 오유를 보여주지 않는다.
-help : 도움말을 보여준다.
-R, -ecursive: 하위등록부에 있는 모든 등록부/파일에 대해 적용변경한다.
-reference=filename : 지정한 파일을 참조하여 권한을 변경한다.
-v, -verbose : 각 파일에 대해 변경에 대한 정보나 변경되지 않은 정보를 상세히 보여준다.
-version : 판본정보를 보여준다.

chmod명령문은 파일이나 파일등록부를 새로운 권한으로 변경하는 명령문이다.

파일의 소유자나 체계관리자만이 chmod를 사용할수 있으며 파일의 소유자, 파일의 집단, 다른 사용자로 나누어 설정할수 있다.

chmod를 리용한 파일 및 파일등록부의 권한변경에는 8진수를 리용한 변경법과 기호에 의한 변경법이 있다. 방법이 달라도 같은 변경된 결과를 얻을수 있다.

권한보기

파일이나 등록부는 모두 매개의 권한을 가지고있다.

파일이나 등록부의 권한은 ls명령을 리용하여 볼수 있다.

출력된 내용을 다음과 같이 할수 있다.

```
kcc@linux:~
  권한        파일소유자  파일집단  용량   파일생성시간     파일명
drwxrwxrw-   1  first    paduk    46   JAN 21  18:23   dir
-rw-r--x     1  first    paduk    46   JAN 21  18:23   test.txt
```

《권한》은 파일을 사용할수 있는 상태를 의미한다.

권 한	설 명	8진수표현
d-------	파일/등록부의 구분(파일:-, 등록부:d)	
-r-------	파일/등록부의 소유자가 읽기권한이 있다.	400
--w------	파일/등록부의 소유자가 쓰기권한이 있다.	200
---x------	파일/등록부의 소유자가 실행권한이 있다.	100
----r-----	파일/등록부의 집단이 읽기권한이 있다.	40
-----w----	파일/등록부의 집단이 쓰기권한이 있다.	20
------x---	파일/등록부의 집단이 실행권한이 있다.	10
-------r--	다른 사용자가 읽기권한이 있다.	4
--------w-	다른 사용자가 쓰기권한이 있다.	2
---------x	다른 사용자가 실행권한이 있다.	1

기호에 의한 파일의 권한변경

기호에 의한 변경방법은 주요한 기능을 기호화한 조합으로 설정할수 있다.

사용자기호는 파일 및 등록부를 사용하려는 사용자에 따른 구분이다. 파일 및 등록부에 대해 소유자나 집단외의 사용자에게 다른 권한을 줄수 있다. 설정하려는 권한을 권한기호에서 정하여 설정기호를 리용해서 조합하여 사용한다. 실례로 test.txt라는 파일을 다른 사용자가 읽을수 있는 권한을 주는 경우에 다음과 같이 입력한다.

```
# chmod o+r test.txt
```

test.txt파일에 한해서 소유자와 집단에게 읽기쓰기권한을 주고 다른 사용자의 읽기권한을 금지하려는 경우에 다음과 같이 입력한다.

```
# chmod ug+rw, o-r test.txt
```

이 경우에 반점(,)을 리용하여 한번에 중복설정할수 있다.

사용자기호

U	user	파일/등록부의 소유자
G	group	파일/등록부의 집단
O	other	다른 사용자
a	all	소유자, 집단, 다른 사용자 모두(아무런 표시도 안하는 경우, 표준적으로 설정됨)

[주의]
웨브페지는 기본적으로 644상태이다. 그것은 웨브봉사기데몬의 소유자(nobody)는 other이기때문이다.

권한기호

r	read	파일 및 파일등록부에 읽기권한을 준다.
w	write	파일 및 파일등록부에 쓰기권한을 준다.
x	execute	파일 및 파일등록부에 실행권한을 준다.
s	set user(group) ID	파일실행시에 파일의 소유자 혹은 집단권한으로 실행된다.
t	sticky bit	Stickyt비트를 설정한다.
u	user	현재 소유자의 권한설정과 같은 내용으로 변경된다.
g	gruop	현재 집단의 권한설정과 같은 내용으로 변경된다.
o	other	현재 다른 사용자의 권한설정과 같은 내용으로 변경된다.
l	locking	강제로 파일에 열쇠를 건다.

설정기호

+	권한허가	지정한 권한을 허가한다.
−	권한금지	지정된 권한을 금지시킨다.
=	권한지정	지정된 권한만 허가하고 나머지는 금지시킨다.

8진수를 리용한 권한변경

4자리의 8진수를 리용하여 파일/등록부의 권한을 변경한다. 아래의 8진수표에 있는 수자를 더해서 사용한다.

실례로 test.txt라는 파일에 대해 소유자에게는 읽기쓰기실행권한을, 집단에게는 읽기실행권한을, 다른 사용자에게는 읽기권한을 부여하려는 경우에 다음과 같이 입력한다.

```
# chmod 754 test.txt
```

파일소유자위치에 읽기(4), 쓰기(2), 실행(1) 등을 더해서 7을 넣는다. 그리고 집단위치에는 읽기(4), 실행(1)을 더해서 5를, 다른 사용자위치에는 읽기권한만 주어 4를 넣어준다.

8진수방식의 변경

파일/등록부의 소유자

0400	파일/등록부의 소유자에게 읽기권한을 준다.
0200	파일/등록부의 소유자에게 쓰기권한을 준다.
0100	파일/등록부의 소유자에게 실행권한을 준다.

파일/등록부의 집단

0040	파일/등록부의 집단에게 읽기 권한을 준다.
0020	파일/등록부의 집단에게 쓰기 권한을 준다.
0010	파일/등록부의 집단에게 실행 권한을 준다.

다른 사용자

0004	다른 사용자에게 읽기 권한을 준다.
0002	다른 사용자에게 쓰기 권한을 준다.
0001	다른 사용자에게 실행 권한을 준다.

4000	파일실행시에 파일의 소유자 혹은 집단 권한으로 실행된다.
2000	파일실행시에 파일의 소유자 혹은 집단 권한으로 실행된다.
1000	Sticky비트

파일정보를 보기 위해 ls를 사용한다.

```
kcc@linux:~

# ls -al
drwxr-xr-x   5  root  root  4096      1월 26일 10:52.
drwxr-xr-x   5  root  root  4096      1월 26일 10:53..
-rwxr-xr-x   5  root  root  3777225   1월 30일 10:53 test1.bmp
drwxr-xr-x   5  root  fish  4096      3월 1일10:53 first
```

[주의]

SetUID(4000), SetGID(2000) 실행파일의 내부에 쉘을 실행시킬수 있는 코드가 있는 경우 보안에 미약하다. 따라서 꼭 필요한 경우가 아니라면 사용하지 않는것이 안전하다.

test1.bmp파일에 대해 소유자는 《읽기쓰기실행》을, 집단은 《읽기쓰기》를, 다른 사용자는 《실행》 등의 권한을 가지도록 변경해보자.

기호에 의한 권한변경의 실례

현재 소유자는 읽기 쓰기 실행권한으로 되여있으므로 수정할 필요가 없고 집단은 읽기 실행권한으로 되여있으므로 실행권한을 금지시키고 쓰기권한을 준다. 그리고 다른 사용자는 읽기 실행권한으로 되여있으므로 읽기권한을 금지시키면 된다.

집단(u)에게 실행(x)권한금지, 쓰기(w)권한을 주기: u－x＋w

사용자(o)에게 읽기(r)권한금지: o－r

```
# chmod u－x＋w, o－r test1.bmp
```

8진수를 리용한 권한변경의 실례

소유자에게는 읽기(400) 쓰기(200) 실행(100)권한을, 집단에는 읽기(40) 쓰기(20)를, 다른 사용자에게는 실행(1)권한을 준다.

소유자 읽기 쓰기 실행: 400＋200＋100: 700

집단 읽기 쓰기: 40＋20 : 60

다른 사용자 실행: 1: 1

```
# chmod 761 test1.bmp
```

【관련명령문】

chfn : 사용자정보변경
chgrp : 집단변경
chown : 소유자변경
chsh : 쉘변경

참 고

SetUID와 SetGID

파일이나 등록부의 파일형식과 권한에 대한 내용은 해당 파일의 inode에 16bit로 보관되여있다. 그중 1~12는 아래에서 보는바와 같이 파일의 실행권한에 대한것이다.

파일ID bit	파일실행 bit	소유자권한 bit	집단권한 bit	기타 권한 bit
16 15 14 13	12 11 10	9 8 7 6	6 5 4	3 2 1

그중 10~12bit는 실행파일을 어떤 방식으로 실행시킬것인가의 내용을 정의하는 부분이다. 이 부분의 형식은 앞에서 언급한 setuid(4000), setgid(2000), sticky(1000) bit이다. 이 bit는 해당 파일의 속성이 등록부인 경우에 무시된다.

한편 UID와 GID는 다음과 같이 설명할수 있다.

- RUID, RGID: 실제(really) 소유자와 집단, 실제 사용자를 검사하며 그 값은 login할 때에 얻는다.
- EUID, EGID, SGID: 파일의 접근권한을 검사확인하는데 사용한다.

조작체계에서 이 파일이 실행된 프로쎄스가 다른 파일에 접근할 때 사용하는 권한이다. 보안과 관련있다.

실행파일이 실행되면 주기억에 적재되여 프로쎄스가 생성된다. 이 실행파일을 소유자외의 사람이 실행하다가 보관해야 하는 경우를 처리하게 위해 프로쎄스와 파일간에 권한을 검사확인할수 있는 통로로 사용하는것이 EUID/EGID이다. 어떤 프로

그람을 실행할 때 프로세스의 EUID/EGID는 대개 RUID/RGID이다. 그러나 프로그람에 따라서는 이 파일이 실행될 때 프로세스의 EUID가 파일소유자가 되도록 설정하거나 EGID가 파일집단이 되도록 설정할수도 있다. 례하면 passwd와 같은 프로그람의 경우, 어떤 사람이 실행하여도 프로세스로서 실행되는동안 이 프로세스는 관리자의 권한으로 동작한다. 즉 이 프로세스가 실행되여 전달하는 모든 명령과 권한은 관리자가 사용하는것과 같다.

이때 프로그람의 내부에 쉘을 실행시키는 코드가 있다면 이와 같이 만들어지는 쉘은 관리자의 쉘로 된다. 그러므로 함부로 이와 같은 권한을 사용하는것은 아주 위험하다. 자세한 내용에 대해서는 보안, 해킹관련의 서적을 참고하도록 한다.

chown

/bin/chown

소유자변경

파일에 대한 사용자와 집단을 변경하는 명령문이다.

형식 : chown[설정값…]소유자:[집단]파일...
chown[설정값...].집단파일...

-f, -silent, -qufet : chown지령으로 변경되지 않은 파일들에 대해 오유통보문을 보여주지 않는다.

-R, -recursive : 하위의 등록부/파일에 모두 적용하여 바꾼다.

-v, -verbose : 변경진행에 대해 자세하게 상태를 보여준다.

-help : 도움말을 보여준다.

-version : 명령문에 대한 판본을 보여준다.

chown명령문은 파일에 대한 소유자와 집단을 지정한 소유자의 집단으로 변경한다.

Apache웨브봉사기봉사를 위한 index등록부를 한번 보자.

Red Hat패키쥐로 설치한 경우, 이러한 등록부로는 /var/www/html이 된다. 현재index.hml의 소유자와 집단은 모두 admin으로 되여있다.

```
kcc@linux:~
# ls -al /var/www/html/index.html
-rw-r--r--  1 admin admin 408  12월 3 09:34 /var/www/htm/index.html
```

index.html의 소유자와 집단을 모두 webmaster로 변경해보자.

```
# chown webmaster:webmaster /var/www/html/index.html
```

index.html파일의 소유자와 집단이 모두 webmaster로 변경되였다.

```
kcc@linux:~

# ls -al /var/www/html/index.html
-rw-r—r— 1 webmaster webmaster 408 12월 3 09:36/var/www/html/
index.html
```

소유자만 변경하는 경우에 다음과 같이 한다.

```
# chown admin index.html
```

변경된 파일의 소유권을 확인해보자. index.html파일의 소유자는 admin이 되였으며 집단은 webmaster집단이 되였다.

```
kcc@linux:~

# ls -al /var/www/html/index.html
-rw-r—r— 1 admin webmaste 408 12월 3 09:36/var/www/html/index.html
```

집단만 변경하는 경우도 있는데 이때에는 점(.)표기로 시작하는 집단을 지정한다.

```
# chown .webmine index.html
```

[주의]
chgrp명령으로도 같은 작업을 수행할수 있다.

변경된 집단을 확인해보자. 소유자는 admin그대로 유지되여있으며 변경된 webmin집단을 확인할수 있다.

```
kcc@linux:~

# ls -al /var/www/html/index.html
-rw-r—r— 1 admin webmin 408 12월 3 09:36/var/www/html/index.html
```

/var/www/html을 포함하여 아래준위 파일등록부 및 파일까지 모두 소유권을 적용 변경하려면 -R설정값을 사용한다.

현재 /var/www/html등록부내의 소유권은 admin식별이름과 집단으로 되여있다.

```
kcc@linux:~

# ls -al /var/www/html
합계 28
drwxr-xr-x  5 admin     admin     4096 2월 4 15:06.
drwxr-xr-x  9 admin     admin     4096 5월 2 15:11..
lrwxrwxrwx  1 admin     admin     24 12월 14 19:52 htdig ->
../../../usr/share/htdig
-rw-r—r—    1 webmaster webmaster 408 12월 3 09:34 index.html
drwxr-xr-x  9 admin     admin     4096  2월 4 15:06 manual
```

/var/www/html등록부를 포함하여 그 아래준위등록부의 파일과 등록부까지 webmaster식별이름과 집단으로 변경해보자. 이때에는 -R설정값을 사용한다.

```
kcc@linux:~

# chown -R webmaster:webmaster /var/www/html
합계 28
drwxr-xr-x  5 webmaster webmaster 4096 2월 4 15:06.
drwxr-xr-x  9 admin     admin     4096 5월 2 15:11..
lrwxrwxrwx  1 webmaster webmaster 24 12월 14 19:52 htdig ->
../../../usr/share/htdig
-rw-r—r—    1 webmaster webmaster 408 12월 3 09:34 index.html
drwxr-xr-x  9 webmaster webmaster 4096  2월 4 15:06 manual
```

【관련명령문】

newgrp : 현재 속해있는 집단을 변경시킨다.

chroot /usr/sbin/chroot

가상뿌리등록부

보안목적으로 새로운 가상 뿌리등록부를 생성하여 원격봉사로 접속하는 경우에 이 파일등록부의 웃준위에로의 이동을 불가능하게 하는 명령문이다.

형식 : chroot 새로운 뿌리경로[명령문]
chroot [설정값]

-help : 도움말을 보여준다.
-version : 명령문에 대한 판본을 보여준다.

사용자가 시도하는 명령문은 보통 기본 뿌리등록부인 /를 기준으로 한 $PATH에 대해 명령문을 해석한다.

chroot는 뿌리등록부(root directory)를 일반적인 환경과 다르게 체계관리자가 지정한 파일등록부로 인식하도록 변경시키게 된다.

이것은 체계관리나 보안상의 측면으로 볼수 있다. 보안상의 측면에서 보면 외부침입자가 원격체계에 대해 쉘식별이름을 얻어 뿌리쉘(root shell)을 따내기 위한 도구를 시도하여도 제대로 될수 없으므로 보다 안전한 체계를 만들수 있는것이다.

실례를 들면 웨브봉사기의 기본 파일등록부인 /var/www/html아래에 체계의 /인것처럼 /파일등록부에 련관된 기본 파일등록부를 만들고 chroot를 설정하여 침해자가 /인것처럼 속이는 역할을 한다. 이 설정에 관한 자세한 내용에 대해서는 다른 지도서를 참고하기 바란다.

[주의]
이 명령을 사용할 때에는 봉사를 사용하는데 필요한 서고와 실행파일을 모두 이 경로에 복사해야 한다. 그렇지 않은 경우에는 봉사가 필요한 내용을 찾지 못한다.

chsh

/usr/bin/chsh

쉘변경

체계가입하여 사용하는 기본쉘에서 림시로 다른 쉘을 사용하게 하는 명령문이다.

형식 : chsh [설정값] [식별이름]

-s, -shell : 지정하는 쉘을 앞으로 사용할 log in 쉘로 바꾼다.
-l, -list-shells : /etc/shells 파일안에 지정된 쉘을 라렬하고 완료한다.
-u, -help : 도움말을 보여준다.
-v, -version : 판번호정보를 보여주고 완료한다.

chsh명령문은 체계가입된 쉘을 변경한다. 이것은 /etc/shells에 등록된 쉘만을 변경할수 있다. 만일 /etc/shells파일에 존재하지 않는 쉘은 오유통보문을 보여주며 쉘의 지정은 전체를 지정해야 한다.
먼저 지금 체계에서 사용할수 있는 쉘을 보자.

```
kcc@linux:~
# cat /etc/shells
/bin/sh
/bin/bash
/bin/bash2
/bin/ash
/bin/bsh
/bin/tcsh
/bin/csh
/bin/ksh
/bin/zsh
```

chsh -l은 /etc/shells의 파일내용과 같은 내용을 보여준다.

```
kcc@linux:~

# chsh -l
/bin/sh
/bin/bash
/bin/bash2
/bin/ash
/bin/bsh
/bin/tcsh
/bin/csh
/bin/ksh
/bin/zsh
```

그러면 chsh명령문을 사용하여 /bin/csh로 변경해보자.

```
kcc@linux:~

# chsh
Changing shell for root.
New shell [/bin/bash]: /bin/csh
Shell changed.
```

-s명령을 사용하여 직접 지정한 쉘로 변경할수도 있다.

```
kcc@linux:~

# chsh -s /bin/bash
Changing shell for root.
Shell changed.
```

자기가 사용하는 쉘을 알아보기 위해서는 다음과 같이 한다.

```
# echo $ SHELL
```

참 고

쉘의 종류

쉘은 분기(fork), 체계호출, 열기, 읽기, 쓰기, 닫기라는 기본기능으로 명령문을 해석하여 프로쎄스(Process)를 생성하고 핵심부(Kernel)와 사용자사이를 련결하는 도구이다.

Linux에서 사용할수 있는 쉘은 다양한데 그중 대표적인것은 다음과 같다.

- Bourne shell (sh) : 벨연구소에서 만들어낸것으로서 제일 초기에 사용된 쉘
- C shell (csh) : 버클리대학에서 만든것으로서 C와 류사한 쉘
- TC shell(tcsh): Freeware로 C쉘의 모든 기능을 제공
- Kom shell(ksh): 벨연구소의 다비드 컨(David Korn)에 의해 만들어진 UNIX표준의 쉘
- Bourne Again SHell (bash) : Linux의 개발자인 토발즈 리누스가 Linux 최초의 핵심부(Kernel)를 위해 Bourne shell을 새롭게 만든것으로서 FSF(Free Software Foundation: 자유쏘프트웨어재단)에 의해 갱신되였다. 최종목적은 IEEE Posix Shell을 대신하는것이다.

	Bourn	C	TC	Korn	BASH
명령리력 (command history)	no	yes	yes	yes	yes
명령문을 간단히 사용하기(alias)	no	yes	yes	yes	yes
쉘스크립트	yes	yes	yes	yes	yes
파일이름완성	no	yes	yes	yes	yes
명령행편집	no	No	yes	yes	no
일감조종 (Job control)	no	yes	yes	yes	yes

clear

/usr/bin/cleat

화면지우기

조종탁(Console)의 본문화면을 지우고 유표를 맨 웃행으로 올린다.
MS-DOS의 cls와 같은 기능이다.

형식 : clear

clear명령문은 DOS의 cls명령과 동일하며 조종탁의 본문화면을 깨끗이 지운다. 이 명령문은 화면의 내용을 모두 지우며 입력독촉상태(Prompt)와 유표(Cursor)는 화면의 제일 우의 왼쪽에 위치한다.

사용법은 간단한다. 아래와 같이 파일목록보기명령문인 ls -a로 목록을 출력하고 화면상에 가득찬 내용을 clear명령을 사용하여 깨끗이 지워보자.

```
kcc@linux:~
# ls -a
.                                       312.jpg
..                                      7monthypaper.pdf
.DCOPserver_ebook.hancom.com            Desktop
.DCOPserver_ebook.hancom.com_:0         Mail
.ICEauthority                           UpdateIP
.MCOP-random-seed                       UpdateIP2
.RealNetworks_RealMediaSDK_60           Word5Patch.sh
.RealNetworks_RealPlayer_60             Word5Patch.sh.lgz
.RealNetworks_RealShared_00             Work
.Xauthority                             anacanda-ks.cfg
# clear
```

cmp

/usr/bin/cmp

파일비교

diff와 같은 기능이지만 보다 간단한 명령이다. 두 파일을 비교하여 다른 부분을 알려준다.

형식: cmp[설정값]

-l, -verbose : 파일들의 차이점을 상세한 내용으로 보여준다.
-s, -quiet, -silent : 아무러한 통보문도 보여주지 않는다.

cmp명령문은 diff명령문을 간단화한것이다. diff가 두 파일간의 차이점을 보여주는 반면에 cmp는 다만 두 파일사이에 서로 다른 부분만을 보여준다.

```
kcc@linux:~
#cmp query query2
query query2 differ: char 8, line1
```

query파일과 query2 파일을 비교해보자. 두 파일은 첫행에서 하나의 글자만이 차이난다.

col /usr/bin/col

개행문자변환

본문파일의 개행문자(LF)와 공백문자(Space) 등을 변환하여 문서속성을 바꾸어준다.

형식 : col[설정값]

-b : 어떠한 BS(backspace:뒤로 지우기)문자도 출력하지 않는다. 이것은 BS문자와 련결되는 마지막 문자만을 출력한다.

-h : 여러 공백문자를 Tab문자로 바꾼다.

-x : 일반적으로는 공백문자를 Tab로 바꾸지만 이 설정값은 여러 공백문자들을 그대로 둔다.

-l 수자 : 지정한 수자값의 수를 주기억에 한번에 둘수 있는 최대행수로 한다. 초기값은 128행이다.

col려파기(Filter)는 \n\r 문자를 \n 문자로 바꾸기, 공백문자를 Tab문자로 바꾸기, 그리고 BS문자를 없애는 기능을 한다.

문서페지(man페지)를 하나의 입력으로 받아 파일로 보관할수 있게 한다.

```
# man httpd col>httph.man
```

colcrt

/usr/bin/colcrt

밑줄(_)문자변환

밑줄문자를 감추거나 다음 행에 풀이표(-)속성을 주는 변환려파기(Filter)이다.

형식 : colcrt[설정값][파일]

- : 밑줄속성이 있는 문자렬을 보여주지 않는다.
-2 : 인쇄할 때에 행간격에서 이상한 오유가 발생하기때문에 밑줄속성이 있는 행에는 다음 행에 풀이표(-)속성을 부여하고 없는 행에는 공백행을 추가한다.

colcrt는 밑줄(_)속성을 풀이표(-)속성으로 바꾸어 주거나 밑줄속성을 보이지 않게 할수 있다. 실례를 통해 보도록 하자.

현재 query라는 파일에는 밑줄문자가 3개 포함되여있는데 date_format, cst_users, cst_productrege 등의 문자렬이다.

```
kcc@linux:~

#cat query
select date_format(uregdate, '%Y%m') date, count(uid) from
cst_users u,
cst_productregs p, cst-productregdetail d
where u.uno=p.preguno
and p.pregno=d.preggno
group by date
```

colcrt는 밑줄문자가 하나의 행아래(-)로 표시된것들을 볼수 있게 한다.

```
kcc@linux:~

# cat query | colcrt
select date format(ureggdate, '%Y%m') date, count(uid) from cst users c, cst
productregs p, cst-productregdetail d
    -                                               -          -
```

```
where u.uno=p.preguno
and p.pregno=d.preggno
```

colcrt -는 밑줄문자를 보이지 않게 하여준다.

```
kcc@linux:~

#cat query | colcrt -
select date format(uregdate, '%Y%m') date, count(uid) from cst users u, cst
productregs p, cst-productregdetail d
where u.uno=p.preguno
and p.pregno=d.preggno
group by date
```

colcrt -2 는 밑줄속성이 있는 행에는 다음행에 (-)속성을 부여하였고 없는 행에는 공백행을 추가한것을 볼수 있다.

```
kcc@linux:~

# cat query | colcrt -2
select date format(uregdate, '%Y%m') date, count(uid) from cst users u, cst

productregs p, cst-productregdetail d
            -                                        -      -
where u.uno=p.preguno

and p.pregno=d.preggno

group by date
```

colrm

/usr/bin/colrm

특정렬의 삭제

파일에서 선택된 렬(column)을 삭제하는 려과기(filter)이다.

형식 : colrm [시작렬[완료렬]

colrm명령문의 첨수가 하나의 수자인 경우, 지정한 수자의 시작부터 끝까지의 행을 삭제한다. 만일 시작렬과 완료렬을 지정하면 지정한 시작렬과 완료렬까지의 행을 삭제한다.

실례를 통해 보기로 하자.

uname명령은 체계정보를 출력하게 된다. 그중 첫번째는 체계의 형식을, 두번째는 체계의 호스트이름을 출력한다. calrm명령을 리용하여 uname으로 볼수 있는 체계 호스트이름만을 뽑아내보자.

uname에 의해 나온 호스트명은 ns.linuxroot.co.kp이다,

```
kcc@linux:~
# uname -a
Linux ns.linuxroot.co.kp 2.4.13-1hl #1 2001. 11. 04(일) 04:04:58 KST i686
unknown
```

웃행의 체계형식을 colrm 1 6 명령으로 삭제해보자. 여기에서 1은 시작렬을, 6은 완료렬을 의미한다.

```
kcc@linux:~
# uname -a | colrm 1 6
ns.linuxroot.co.kp 2.4.13-1hl #1 2001. 11. 04(일) 04:04:58 KST i686
unknown
```

호스트명다음의 모든 정보를 추출(filtering)하려는 경우, 19렬이후는 모두 삭제하면 된다.

```
kcc@linux:~
# uname -a | colrm 1 6 | colrm 19
ns.linuxroot.co.kp
```

참 고

호스트접속 때에 보여주는 통보문변경

호스트에 접속하는 경우, 대부분이 체계의 핵심부정보와 배포판정보를 보여 주게 되는데 이것은 보안상의 견지에서 약점으로 될수 있다. 이것을 보여주는 파일은 어떤것인가?

/etc/issue파일은 망접속때에 보게 되는 체계통보문을 담고있다.

/etc/issue.net파일은 망접속때에 보여주는 체계통보문과 같은 정보를 원격접속하는 호스트에 보여주게 된다. 이 정보를 보여주려는 정보로 수정하면 된다.

하지만 이것은 /etc/rc.d/rc.local 파일에서 체계기동시에 체계정보를 다시 가져오기때문에 아래와 같이 /etc/issue.net에 대하여 주해처리하도록 하여야 한다.

```
echo "" > /etc/issue
echo "$R" >> /etc/issue
echo "Kernel $(uname -r) on $a $SMP$(uname -m) " >> /etc/issue
# cp -f /etc/issue /etc/issue.net
```

/etc/motd파일은 체계접속 후 보여주는 통보문정보를 담고있다.

column /usr/bin/column

파일정렬

본문파일의 내용을 가로형식으로 보기 좋게 정렬하여 보여주는 명령문이다.

형식 : column[설정값][파일...]

-c num : 전체 가로폭을 columns값으로 하여 정렬한다.

-s char : -t 설정을 위해 가로칸조사에 사용될 문자를 지정한다.

-t : 입력되는 내용의 가로칸개수를 조사하고 그것을 바탕으로 출력양식을 정한다. 입력되는 내용의 가로칸개수는 공백문자로 구분한다.

-x : 가로방향으로 먼저 라렬하고 다음에 세로방향으로 라렬한다.

column명령문은 렬을 형식화하는 명령문이다. 먼저 세로방향으로 보여주고 그 다음에 가로방향으로 보여준다. 본문의 결과에 적당한 Tab식별자를 넣어 보기 좋게 한다. 아래의것은 ls -l 명령으로 출력되여 나오는 현재 파일등록부안의 파일목록을 리해하기 쉽도록 하기 위해 렬제목을 붙여준 형태이다. Sed 1d명령으로 합계 부분을 모두 삭제하고 printf 명령으로 개개의 항목(Field)에 대한 렬제목을 아래와 같이 지정하였다. 이것을 column -t로 받아 입력되는 내용의 가로칸개수를 조사하여 출력양식을 지정하도록 하였다.

```
kcc@linux:~
# (print "PERM LINKS OWNER GROUP SIZE MONTH DAY HH:MM/ YE
AR NAME\n" ;ls -l | sed 1d) column -t
PERM          LINKS  OWNER  GROUP  SIZE  MONTH    DAY
HH:MM/YEAR  NAME
-rw-r-r- 1     root    root    798720   8월   5   10:04
Command.hwp
-rw-r-r- 1     root    root    321475   6월   14  11:06
20040608.hwp
-rw-r-r- 1     root    root    384283   6월   24  00:18
20040620.hwp
-rw-r-r- 1     root    root    23553    6월   19  16:59
Planning.hwp
```

comm

/usr/bin/comm

한행씩 비교

정렬된 두 파일을 한행씩 비교하는 명령이다.

형식 : comm[설정값]파일 1 파일 2

-1 : 파일2를 기준으로 파일1과 비교하여 차이나는 부분을 보여주고 다음 렬에 같은 부분을 보여준다.
-2 : 파일1을 기준으로 파일2와 비교하여 같지 않은 부분을 보여주고 다음 렬에 같은 부분을 보여준다.
-3 : 파일1과 파일2를 비교하여 첫번째 렬에 파일1에만 있는 부분을, 두번째 렬에 파일2에만 있는 내용을 보여준다.
-help : 도움말을 보여준다.

comm명령문은 다목적의 파일비교편의프로그람으로서 이 명령을 수행하기 전에 비교할 파일이 정렬되여있어야 정확하게 결과를 얻을수 있다. 실례를 통해 보기로 하자. 먼저 test1과 test2 파일을 sort명령으로 정렬하자.

```
kcc@linux:~
# sort <test1> test1
# cat test1
1
3
5
6
7
# sort <test2> test2
# cat comm2
2
4
6
7
8
9
```

아래와 같이 -1로 하여주면 파일2를 기준으로 하여 파일1과 같지 않은 부분은 첫번째 렬에, 같은 내용은 2번째 렬에 보여준다.

```
kcc@linux:~
# comm -1 comm1 comm2
2
4
        6
        7
8
9
```

아래와 같이 -2로 하여주면 파일1을 기준으로 하여 첫번째 렬에 파일2와 같지 않은 부분을, 2번째 렬에 같은 내용을 보여준다.

```
kcc@linux:~
# comm -2 comm1 comm2
1
3
5
        6
        7
```

-12로 주면 test1파일과 test2파일의 공통된 내용을 보여주는데 이것을 uniq명령에 의해 렬에 중복되지 않는 내용으로 test3파일을 생성한다.

```
# comm -12 test1 test2 | uniq > test3
```

compress /usr/bin/compress

압축파일

파일을 .Z의 형태로 압축하는 명령문이다.

형식 : compress[설정값]파일...

-b maxbits: 최대비트수를 제한한다.
-c : 기본생성파일인 .Z의 형태가 아닌 지정한 파일로 생성한다.
-d : 압축된 파일의 압축을 해제한다. 이것은 uncompress 지령과 같다.
-f : 이미 이전에 압축파일이 존재하여도 그를 무시하고 압축파일을 생성한다.
-r : 지정한것이 등록부인 경우, 하위등록부와 파일까지 모두 압축한다.
-v : 압축통계를 보여준다.
-V : 판본과 명령문에 대한 정보를 보여준다.

compress명령문은 하나이상의 파일을 파일이름에 Z확장자를 붙여 새로운 파일압축을 생성하게 한다.
다음과 같은 파일들이 있다고 가정하자.

```
kcc@linux:~
# ls
20040417_Command.hwp 20040608.hwp 20040612titlesample1.hwp
20040620.hwp Planning.hwp
```

먼저 이 파일들을 tar명령의 -cf설정에 의해 모두 하나의 파일로 묶어보자.

```
# tar -cf test.tar*
```

test.tar파일로 파일이 묶여진것을 볼수 있다.

```
kcc@linux:~
# ls test.tar
test.tar
```

파일압축

파일을 compress -f명령으로 압축할수 있다. compress압축은 기본적으로 .Z확장자를 가진다. 기본적으로 원본파일은 지워지고 압축된 파일만 남는다. -v를 사용하여 압축률과 압축과정을 확인할수 있다.

```
kcc@linux:~
# compress -f -v test.tar
sum.awk: — replaced with test.tar.Z Compression: 28.07%

# ls test.tar*
test.tar.Z
```

파일압축의 해제

.Z확장자를 가진 test.tar.Z파일은 uncompress명령으로 압축을 해제한다. -v를 사용하여 압축해제결과를 볼수 있다.

```
kcc@linux:~
# uncompress -v test.tar.Z
sum.awk.Z: -replaced with sum.awk
```

압축해제된 test.tar파일을 tar -xf명령에 의해 파일묶음해제를 할수 있다.

```
# tar xf test.tar
```

cp

/bin/cp

파일복사

파일을 지정된 경로에 복사하는 명령문이다.

형식 : cp[설정값]원본파일 복사파일, cp[설정값]파일...등록부

-a, -archive : 원본파일의 속성, 련결(link)정보들을 그대로 유지하면서 복사한다. 이 설정값은 -dpR설정값과 같은 역할을 한다.
-b. -backup : 복사대상파일이 있는 경우에 예비보관파일을 만든다.
-d, -no-deference : 원본파일이 쏘프트련결파일이면 쏘프트련결원본을 복사한다.
-f, -force : 복사대상파일이 있는 경우에 복사대상파일을 강제로 지우고 복사한다.
-i, -interactive : 복사대상파일이 있는 경우에 사용자에게 복사에 대한 여부를 문의한다.
-l, -link : 기호련결형식으로 복사한다. 물론 하드련결형식이므로 경로는 복사할수 없다.
-p, -panets: 원본파일에 경로와 함께 지정한 경우에 그 경로대로 복사한다.
-p, -preserver: 원본파일의 소유자, 집단, 권한, 시간정보들을 그대로 보관하여 복사한다.
-r : 일반파일이면 그냥 복사하고 만약 원본이 경로이면 그 경로와 함께 경로안에 있는 모든 하위경로와 파일들을 복사한다.
-s, -symbolic-link : 경로가 아닌 일반 파일을 기호련결형식으로 복사한다.
-u, -pdate : 복사할 대상이 이미 있는데 이 파일의 변경날자가 같거나 더 최근의것이면 복사하지 않는다.
-v, -verbose : 복사상태를 상세히 보여준다.
-x, -one-file-system : 원본과 대상파일의 파일체계가 다른 경우에는 복사하지 않는다.
-R, -recusijve : 등록부를 복사하는 경우, 하위등록부와 파일을 모두 복사한다.
-help : 도움말을 보여준다.
-version : 판본정보를 보여준다.

cp는 파일을 또 하나의 파일이나 다른 등록부에로 복사하는 명령문이다.

복사할 때에 대상파일이 있는 경우 사용자에게 문의함이 없이 복사한다.

만일 이 복사가 잘못된 경우에 이전의 정보우에 다른 내용이 덧씌워진 파일을 복구하는것은 거의나 불가능하다. 이러한 실수를 방지하기 위해 alias설정으로 -i를 설정하였다.

```
kcc@linux:~
# alias | grep cp
alias cp='cp -i'
```

아래에서 Apache설정파일과 이름봉사기설정파일을 home등록부에 복사한다.

```
#cp /etc/httpd.conf etc/named.conf ~
```

-R설정값을 주면 웨브봉사기의 기본등록부인 /var/www/html과 모든 아래준위등록부 및 파일을 /backup등록부에로 복사한다.

```
#cp -R /var/www/html /backup
```

참　고

inode

Linux나 UNIX에서 파일을 나타내는것은 파일이름이 아니며 이 파일체계에서 어떤 파일을 인식하는것은 inode라는 수자이다. 간단히 말해서 파일이름은 inode구조체표에 보관된 파일정보를 사람이 인식하기 쉽게 만든것에 지나지 않는다. 여기에서는 흥미있는 현상을 확인할수 있는데 다음과 같은 경우를 실험해보기 바란다.

먼저 적당한 이름의 빈 파일을 하나 만들고 inode를 확인해보자. 잊어버릴수도 있으므로 기록하여놓고 다음에로 넘어가자. 이 파일을 mv명령을 사용하여 다른 이름으로 바꾸고 다시 inode를 확인해보면 이 2개의 파일은 서로 이름이 다르지만 같은 inode라는것을 알수 있다. 이것은 우리가 어떤 이름을 사용하든지간에 체계에서는 같은 파일로 인식하고있다는것을 보여준다.

그러면 이번에는 cp명령을 사용하여 원래의 이름으로 파일을 복사하고 다시 비교해보자. 이때에는 inode가 달라졌음을 알수 있는데 이것은 같은 이름이였다고 해도 체계는 서로 다른 두 파일로 인식한다는것을 보여준다.

cpio

/bin/cpio

사본만들기

파일을 복사하고 압축하는 명령문이다.

형식:

cpio -o, -create [설정값] : 표준입력으로 받는 파일목록을 압축된 파일출력형태로 복사한다.

cpio -i, -extract [설정값] [패턴] : 지정한 패턴의 형태에 맞는 압축파일로부터 입력을 받아 압축을 푼 파일로 복사한다.

cpio -p, -pass-through[설정값] 등록부 : 꼭같은 체계에서 지정된 다른 파일등록부에로 파일을 복사한다.

-0, -null : 파일이름의 끝이 LF(Line Feed)가 아니라 null로 끝난다.

-a, -reset-access-time : 파일접근시간을 재설정한다.

-A, -append: 압축된 파일에 파일을 추가한다. -o설정값에서만 작용한다.

-B : I/O 블로크를 5120 bytes로 지정한다. 초기값은 512bytes이다.

-block-size=BLOCK-SIZE : I/O블로크의 크기를 BLOCK-SIZE *512bytes로 지정한다.

-c : SVR4형식을 사용하기 위해 《-H newc》설정값을 리용한다. 이전의 ASCII형식을 사용하려면 《-H odc》를 리용한다.

-C IO-SIZE, -io-size=IO-SIZE : I/O블로크크기를 지정한 IO-SIZE로 지정한다.

-d, -make-directories : 지정한 등록부를 생성한다.

-E FILE, -pattern-file=FILE : -i설정값에서 사용하며 지정한 패턴의 형식에 맞는 파일을 읽는다.

-f, -nonmatching : 주어진 패턴에 관계없이 파일을 복사한다.

-F, -file=archive : 표준입력이나 표준출력대신에 주어진 파일이름으로 압축한다.

-force-local : -F, -I, -O 설정값과 함께 압축된 파일을 국부파일(Local File)로 인식한다.

-H FORMAT, -format=FORMAT: 주어진 FORMAT형식으로 압축한다.

bin : 2진형식
odc : 이전의 POSIX이식형식
newc : 새로운 SVR4 이식형식
crc : 새로운 SVR4 이식형식에 검사합(Check Sum)을 추가
tar : tar형식
ustar : POSIX.1 tar 형식
hpbin : HPUX의 cpio에 쓰이는 2진형식
hpodc : HPUX의 cpio에 사용되는 이식형식

-I archive : 표준입력대신 archive파일명을 사용한다. 복구할 때에 다중기록권(Multivolume)을 효과적으로 처리할수 있다.

-l, -link : 복사하기보다는 파일을 련결(Link)한다.

-L, -dereference: 련결을 복사하는것이 아니라 원본파일을 복사한다.

-m, -preserve-modificataion-time: 파일을 만들때에 파일수정회수를 같이 보관한다.

-M MESSAGE, -message=MESSAGE : 자기테프, 플로피디스크 등에서 기록매체여벌복사크기(Media Backup Size)에 도달했을 때에 지정한 MESSAGE(통보문)를 보여준 후 입력독촉(Prompt)을 나타내여 새로운것을 삽입하게 한다.

-n, -numeric-uid-gid : 실행 때에 동작되는 상세파일이름을 보여주는 것이 아니라 UID와 GID를 보여준다.

-O archive : 표준출력대신에 archive파일이름을 사용한다. 여벌복사나 파일보관 때에 다중기록권을 효과적으로 처리할수 있다.

-quiet : 복사된 블로크개수를 보여주지 않는다.

-r, -rename : 서로의 대화식으로 파일을 다시 지정한다.

-R [user][:.][group], -owner [user][:.][group] : -o과 -p 설정값에서 지정한 소유권으로 파일을 생성한다.

-t, -list : 입력목록표를 출력한다.

-u, -unconditional: 이전의 파일이 있든, 없든간에 관계없이 무조건 파일을 복구한다.

-v, -verbose : 실행동안 파일들을 보여주거나 -t 설정값과 함께 쓰이여 ls -l star로 목록을 보여준다.

이것은 cpio체계예비보관에 자주 사용되는 명령문이다. 이 명령문의 사용은 복잡한 예비보관의 적용과 융통성을 지원할수 있게 한다.

예비보관장치인 /dev/ftape에서 파일을 복구해보자.

```
# cd /home/
# cpio -ivcdumB < /dev/ftape
```

참　고

웨브봉사기 예비보관

다른 하나의 하드디스크를 Secondary Master에 삽입한 다음 fdisk명령으로 구획(Partition)을 지정한 다음 재기동한다. 그리고 다음과 같이 mount할 등록부를 생성한다. /dev/hdc1구획을 /hdc1 등록부로 mount한다.

```
# mkdir /hdc1
```

```
# mount /dev/hdc1 /hdc1
# cd /home/
# find . -depth -print | cpio -pmdvl /hdc1
```

crontab /usr/bin/crontab

crontab관리

사용자의 개별 crontab파일관리를 위한 명령문이다.

형식 : crontab[-u 사용자 ID]파일, crontab[-u 사용자 ID]{-l| -r|-e}

-l : 현재의 crontab내용을 표준출력으로 보여준다.
-r : 현재의 crontab내용을 지운다.
-e : crontab 파일의 내용을 편집한다. 편집기로는 VISUAL 이나 EDITOR 환경변수의 편집기를 사용한다.

crontab는 지정한 시간에 지정한 작업을 하도록 설정하기 위한 주기적인 작업일정이다. 이에 비해 at명령은 지정한 시간에 한번만 작업을 시행한다.
crontab설정파일은 /etc/crontab이다.

```
kcc@linux:~
# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
02 4*** root run-parts /etc/cron.daily
22 4** 0 root run-parts /etc/cron.weekly
42 4 1** root run-parts /etc/cron.monthly

0-59/5**** root /usr/bin/mrtg /etc/mrtg/mrtg.cfg
```

그러면 뿌리식별이름에 새로운 작업일정을 등록해보자.
먼저 이전에 crontab에 설정되어있는 작업이 있는가를 확인해보자.

등록된 작업은 없다.

```
kcc@linux:~
# crontab -l
no crontab for root
```

-e설정값으로 편집하게 되면 편집기가 실행되어 새로운 cron작업을 등록할수 있다

```
kcc@linux:~
# crontab -e
no crontab for root - using an empty one
```

다음과 같이 입력해보자

```
0 4 *** echo crontab testing
```

보관하고 나오게 되면 아래의 통보문이 출력된다.

```
"crontab.1314" 1L, 31C written
crontab: installing new crontab
```

그러면 등록된 작업을 -l설정을 리용하여 확인해보자.

```
kcc@linux:~
# crontab -l
# Do NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.1314 installed on Sun Jun 30 23:45:32 2004)
# (Cron version -- $ID: crontab.c, v 2.13 1994/01/17 03:20:37 vixie Exp $)
0 4 *** echo crontab testing
```

cron일정에 등록하는 시간지정에 관한 매 항목은 다음과 같다.

분(Minutes)	0-59
시(Hour)	0-23
일(Day of month)	1-31
월(Month)	1-12 or Jan, Feb, Mar_
요일(Day or week)	0-6(0은 Sunday) or Sun, Mon, Tue, ...

참 고

웨브봉사기의 종류

현재 사용하는 웨브봉사기의 종류에는 여러가지가 있지만 대표적으로는 Apache가 가장 많이 사용되고있으며 그 다음으로는 마이크로쏘프트제품인 IIS가 많이 사용되고있다. 현재 Apache가 전체 웨브봉사기의 60%정도이며 IIS가 30%정도이다.

csplit

/usr/bin/csplit

파일분할

지정하는 기준에 준하여 파일을 자동적으로 나누어주는 명령으로서 내용이 긴 문서파일을 효과적으로 나누어 읽을수 있게 한다.

형식 : csplit [설정값]... 파일 인수 ...

- : 표준입력으로부터 입력을 받는다,

-b suffix, —suffix-format=suffix : 새롭게 생성되는 파일의 접미사(suffix)를 지정한다. 보통 파일은 00,01로부터 시작하여 99까지 생성된다. 아래와 같은 형식지정이 가능하다.

- %d: 부호있는 10진수
- %I : %d와 같다.
- %u: 부호없는 10진수
- %o: 8진수
- %x: 16진수
- %X: %x와 같다.

-f prefix, -prefix=prefix : 새롭게 생성되는 파일의 접두사를 지정할수 있다. prefix00부터 prefix99까지가운데서 prefix부분을 지정하는것이다.

-k, -keep-files : 생성도중에 오유가 발생하여도 새롭게 생성된 파일들을 유지한다.

-n, -digits=num : 생성되는 파일의 길이를 지정한다. 기본값은 2자리이다.

-s, -quiet, -silent : 문자개수를 보여주지 않는다.

-z, -elide-empty-files : 빈 파일을 제거한다.

csplit는 파일안의 내용들가운데서 지정하는 기준에 준하여 여러개의 파일로 분리생성하는 기능을 한다. 아래에 《csplit 파일이름》의 다음에 올수 있는 인수들에 대하여 설명하였다.

이 인수는 파일들을 나눌 기준으로 생각하면 된다.

- n: 현재행부터 n번째행까지의 범위를 파일로 생성한다.

정규표현식/[+n/-n]: 현재행부터 정규표현식을 포함하는 행까지 파일을 생성한다. +n이면 정규표현식을 기준으로 n번째 행까지 더 포함하며 -n이면 정규표현식을 기준으로 n번째 행까지를 제외하게 된다.

%정규표현식%/[+n/-n]: 정규표현식의 행앞부분은 보관되지 않는다.

- {n}: n번 반복한다.
- {*}: 입력이 있을 때까지 반복한다.

실례를 통해 보기로 하자. 다음에 볼수 있는 song.txt 파일은 노래가사를 보관해둔것이다. 이 노래를 매 파일에 나누어 보관해보겠다.

```
kcc@linux:~

# cat song.txt
song.1
이국에 들가에 피여난 꽃도
내 나라 꽃보다 곱지못했소
돌아보면 세상은 넓고 넓어도
내 사는 내 나라 제일로 좋아

song.2
벗들이 부어준 한모금 물도
내 고향 샘처럼 달지못했소
돌아보면 세상은 넓고 넓어도
내 사는 내 나라 제일로 좋아

song.3
해와 별 비치여 밝고 정든 곳
내 다시 안길 때 절을 하였소
돌아보면 세상은 넓고 넓어도
내 사는 내 나라 제일로 좋아
```

[주의]

csplit를 실행했을 때 화면에 출력되는 수자는 새로 생성된 파일의 단어수이다.

이 파일을 매 절별로 나누어 보기로 하자.

```
kcc@linux:~

# csplit song.txt %song.1% /song.2/ /song.3/
111
120
111
```

csplit명령에 의해 xx00, xx01, xx02파일이 생성되였다.

```
kcc@linux:~

# cat xx00
song.1
이국에 들가에 피여난 꽃도
내 나라 꽃보다 곱지못했소
돌아보면 세상은 넓고 넓어도
내 사는 내 나라 제일로 좋아

# cat xx01
song.2
벗들이 부어준 한모금 물도
내 고향 샘처럼 달지못했소
돌아보면 세상은 넓고 넓어도
내 사는 내 나라 제일로 좋아

# cat xx02
song.3
해와 별 비치여 밝고 정든 곳
내 다시 안길 때 절을 하였소
돌아보면 세상은 넓고 넓어도
내 사는 내 나라 제일로 좋아
```

새로 생성되는 파일의 수자는 00부터 시작한다는데 주의해야 한다.

cut

/bin/cut

항목의 골라보기

파일에서 요구하는 항목(Field)을 뽑아내는 명령문이다.

형식 : cut[설정값]...[파일]

-b, -bytes=LIST: 바이트단위로 나타낸다.

-c, -characters=LIST: 문자단위로 나타낸다.

-d, -delimiter=DELIM: 항목의 기본식별자는 TAB이지만 이것을 사용하지 않고 지정된 항목식별자를 사용한다.

-f, -filelds=LIST: 지정한 항목만을 보여준다.

-s, -only-delimited: 항목식별자에 포함되지 않는 행은 보여주지 않는다.

-output-delimiter=STRING: 출력되는 항목식별자로는 지정된것을 사용한다.

-help: 도움말정보를 보여준다.

cut명령문은 awk에서의 print $N과 비슷하지만 awk보다 더 간단하게 사용될수 있다. -b, -c, -f설정값에서는 다음의 특정한 수자범위를 사용할수 있다.

- N: N번째
- N-: N번째부터 마지막까지의 범위
- N-M: N번째부터 M번째지의 범위
- -M: 첫번째부터 N번째까지의 범위

1. OS와 핵심부판본정보의 얻기

```
kcc@linux:~
# uname -a
Linux ns.linuxroot.co.kp 2.4.13-1hl #1 2001.11.04
(일) 04:04:58 KST i686 unknown

# uname -a | cut -d ' ' -f1,3
Linux 2.4.13-1hl
```

2. mount된 파일체계목록의 얻기

```
kcc@linux:~

# cat /etc/mab | cut -d ' ' -f1,2
/dev/hda8 /
none /proc
usbdevis /proc/bus/usb
/dev/hda1 /boot
none /dev/pts
/dev/hda3 /home
none /dev/shm
/dev/hda2 /tmp
/dev/hda7 /var
/dev/hdb1 /hdb1
/dev/hdb2 /hdb2
/dev/hdc1 /hdc1
```

[주의]

who | cut -d ' ' -f1은 제대로 출력되는데 f2, -f3은 예상하였던 결과와 다르게 나오는 등 출력결과의 빈칸이 Tab처럼 보이지만 실제로는 공백인 경우도 있다.

D

date /bin/date

날자보기

필요에 따라 시간과 날자를 다양한 형식으로 출력하는 명령문이다.

형식 : date[설정값][+형식][날자]

-d, —date date : 지정한 date값으로 시간과 날자를 보여준다.
-t, —file=datebata : 지정한 날자파일에서 각 행에 대한 날자를 보여준다.
-l, —iso-8601[=timespec] : ISO-8601 형식으로 보여준다.
만일 timespec를 지정하면 날자, 시, 분, 초에 대한 값중의 하나를 보여준다.
-r, —reference=file : 지정한 파일이 마지막으로 수정된 날자를 보여준다.
-R, —rfc-822 : RFC-822 날자문자렬로 보여준다.
-s, —set=string : 지정한 string으로 시간을 맞춘다.

date는 시간과 날자를 알려준다. 만일 설정값이 없으면 체계의 현재시간과 날자를 보여주며 +로 시작되는 설정값이 있는 경우에는 지시하는 형태대로 보여주게 된다. 기본출력은 다음과같다.

```
kcc@linux:~
# date
2004.07.01. (월) 20:09:50 KST
```

%j는 현재 체계의 년도에서 오늘의 날자까지 몇일이 지났는가를 알수 있다.

```
kcc@linux:~
# date +%j
182
```

%H는 현재시간을 24시간제로 보여주며 %M은 분을 보여준다.

또한 다음과 같은 방법으로 오늘 날자의 년월일을 형식지정하여 표현할수도 있다.

```
kcc@linux:~
# date +%H%M
2009
# date +%Y-%m-%d
2004-07-10
```

--date설정값으로 현재시간으로부터 15일후의 날자를 알수 있다.

```
kcc@linux:~
# date --date '15 days day'
2004.07.17 (수) 20:18:24 KST
```

현재 시간으로부터 3년 2개월전의 날자를 알아보자.

```
kcc@linux:~
# date --date '3years 2months ago'
2001.05.01 (토) 20:19:48 KST
```

또한 이와 같은 날자의 출력결과를 다양한 형식으로 출력할수 있다. 형식을 다양하게 조합하여 자기만의 출력결과를 만들수도 있다. 또한 쉘스크립트와 cron을 리용하여 예비보관이나 기타 작업을 할 때에 쉘스크립트에서 date명령을 리용하여 작업한 날자와 시간의 이름으로 파일을 만들도록 지정할수도 있다.

%	%문자 그 자체
—	항목을 채우지 않는다(기본적으로 0으로 채운다).
-	항목을 공백으로 채운다.
%a	간략화한 요일이름(Sun_Sat)
%b	간략화한 월이름(Jan_Dec)
%c	날자와 시간(Mon Jun 01 10:42:09 KST 2004)
%d	월 기준 일(01..31)
%h	%b와 같다.
%j	년 기준 일(001...366)
%k	24시간 기준 시간(0...23)
%l	12시간 기준 시간(1...12)
%m	년 기준 월(01...12)
%n	새로운 행을 입력한다.
%p	AM or PM
%r	12시간 표현형식(hh:mm:ss[AM|PM]
%s	1970-01-01 00:00:00 UTC로부터 경과된 초
%t	Tab를 입력한다.
%w	요일의 수자(일요일은 0)
%x	현재의 날자표현식(mm/dd/yy)
%z	RFC 822표현의 수자시간대
%A	완전한 요일표현(Sunday...Saturday)
%B	완전환 월표현(January...December)
%D	%m/%d/%y 날자표현식
%H	24시간 표현형식(00...23)
%I	12시간 표현방식(01...12)
%M	분 표현(00…59)
%S	초 표현(00...59)
%T	%H:%M:%S 형태의 시간표현
%U	년 기준 주일의 수자(일요일이 주일의 시작임, 00...53)
%V	년 기준 주일의 수자(월요일이 주일의 시작임, 01...52)
%W	년 기준 주일의 수자(월요일이 주일의 시작임, 00...53)
%X	정의한 시간의 표현(%H:%M:%S)
%Y	년 표현의 네자리 수자(례: 2004)
Z	시간대의 이름(례: KST)

dd

/bin/dd

블로크단위의 복사

기동디스크나 swap파일을 만드는 등의 특수한 복사를 하는 명령문이다.

형식 : dd[설정값]

-help : 도움말을 보여준다.
-version : 판본정보를 보여준다.
Of=file : 표준출력대신 지정한 파일에 작성한다
if=file : 표준입력대신 지정한 파일에서 불러들인다.
ibs=size : 지정한 크기(size)만큼 읽는다.
obs=size : 지정한 크기만큼 쓴다.
bs=size : 지정한 크기만큼 읽고 쓴다.(ibs, obs값은 무시된다.)
count=n : 입력블로크의 ibs크기만큼 복사한다.
conv=ascii : EBCDIC코드를 ASCII코드로 변환한다.
conv=ebcdic : ASCII코드를 EBCDIC코드로 변환한다.
conv=ibm : ASCII코드를 호환 EBCDIC코드로 변환한다.
conv=swab : 입력되는 두 바이트의 순서를 바꾼다. 입력파일에서 짝이 맞지 않는 하나의 바이트가 남는 경우, 마지막 바이트는 그대로 단순히 복사된다.

dd명령문은 변환과정을 거쳐 파일 혹은 표준입출력을 복사한다.

ASCII-EBCDIC간의 변환, 대소문자간의 변환, 입출력바이트쌍을 바꾸기, 입력파일의 처음 혹은 끝을 건너뛰거나 잘라내여 출력파일을 만드는것과 같은 다양한 변환을 할수 있다. 예비보관에서와 같이 대량의 자료를 복사할 때에도 사용한다. dd는 파일내용을 장치에 그대로 쓴다.

[주의]

Windows에서 기동디스크를 작성할 때에는 rawrite를 사용한다.

1.44인치 기동디스크의 만들기

```
kcc@linux:~

# pwd
/boot
# dd if=kernel-image of=/dev/fd0 bs=1440
```

kernel-image는 /boot파일등록부에 있는 kernel image이름을 지정한다.

512M의 swap파일만들기

```
# dd if=/dev/zero of=/swap bs=1024 count=524288
# mkswap /swap 524288
# sync
# swapon /swap
```

free명령으로 작성된 swap파일을 확인한다,

```
# free
```

【관련명령문】

tar : 빠르고 간단한 파일묶기명령문
cpio : 체계예비보관에 사용하는 명령문
mt : 자기테프의 조작명령문

참 고

lilo에 암호걸기

Linux가 설치된 콤퓨터를 보호하기 위하여 기동적재프로그람인 LILO에 암호를 걸어보자.

/etc/lilo.conf파일을 열어 원하는 image아래에 다음과 같이 추가한다.

password=사용하려는 암호

이때에 password만 지정하면 어떤 기동방법을 사용하든지간에 암호가 걸린다.

즉 init 3 이나 init 5로 기동하는 경우에도 암호를 묻게 되므로 원격기동할 때에는 문제가 생길수 있다. 이것을 방지하려면 암호를 설정한 다음행에 《restricted》라고 입력한 뒤에 보관한다.

debugfs

/sbin/debugfs

파일체계조종

ext2파일체계의 오유수정기로서 파일체계를 직접 확인하며 조종할수 있다.

형식 : debugfs[[-설정값]장치명]

-w : 읽기 및 쓰기 방식으로 파일체계를 연다.

debugfs는 호상대화형의 오유수정기이다. debugfs명령문을 실행한 다음에 다음과 같은 명령문들을 사용할수 있다.

[주의]

이 명령은 파일이 보관된 파일체계를 직접 조종하는것이므로 블로크를 잘못 삭제하는 경우, 해당한 파일을 사용할수 없게 될수도 있으므로 주의하여야 한다.

cat 파일이름	Inode하나의 내용을 표준출력으로 출력표시(dump)한다.
cd 등록부	작업등록부를 변경한다.
chroot 등록부	지정한 inode로 root등록부를 변화시킨다.
close	열려진 파일체계를 닫는다.
clri 파일이름	파일에 해당하는 inode의 내용을 지운다.
dump 파일이름 출력파일이름	Inode의 내용을 파일로 출력표시한다.
expand_dir 파일이름	등록부를 확장한다.
find_free_blok [목표]	목표로부터 시작하여 첫번째의 빈 블로크를 찾아 할당한다.
find_free_inode [등록부] [방식]	빈 inode를 찾아서 할당한다.
Freeb 블로크	블로크를 할당되지 않은것으로 표시한다.
freei 파일이름	파일이름에 해당하는 inode를 해제한다.
help	Debugfs가 사용할수 있는 명령문목록을 보여준다.
check 블로크	블로크를 inode로 변환수행한다.
iname inode	inode에 해당하는 파일이름을 보여준다.
Initialize 장치명 블로크크기	장치명에 ext2파일체계를 생성한다.
kill_file 파일이름	파일을 삭제하고 사용한 블로크를 해제한다.
ln 원본파일 목표파일	련결(link)을 만든다.
ls [경로이름]	ls명령문을 실행시킨다.

modify_inode 파일이름	파일이름에 해당하는 inode의 내용을 수정한다.
mkdir 등록부명	지정한 등록부명으로 등록부를 생성한다.
mknod 파일이름 [p|[[c|b]주번호 종속번호]]	특별한 장치파일을 지정한 주번호(major number)와 종속번호(minor number)로 생성한다.
ncheck inode	Inode에서 이름으로 변환을 수행한다.
open[-w] 장치명	파일체계를 연다.
pwd	현재 작업등록부를 보여준다.
quit	debugfs를 완료한다.
rm 파일이름	지정한 파일을 지운다.
mdir 등록부	지정한 등록부를 삭제한다.
setb 블로크	지정한 블로크를 할당된것으로 표시한다.
seti 파일이름	파일이름에 해당하는 inode를 사용하는것으로 표시한다.
show_super_stats	Super block의 목록을 보여준다.
stat 파일이름	지정한 파일이름으로 inode의 목록을 출력표시한다.
estb 블로크	블로크가 할당되여있다고 표시되여있는가를 검사한다.
testi 파일이름	파일이름에 해당하는 inode가 할당되여있다고 표시되여있는가를 검사한다.
unlink 파일	지정한 파일의 련결(link)을 제거한다.
write 원본파일 대상파일	파일이름을 가지는 파일체계에 하나의 파일을 만든 다음 원본파일의 내용을 대상파일에 복사한다.

참 고

rm명령으로 삭제된 파일을 복구하기

Linux에서는 삭제된 파일은 복구가 거의나 불가능하다. debugfs명령으로 방금 삭제된 파일을 한번 복구해보도록 하자.

> ⚠ **경 고**
>
> 만일 삭제후 새로운 파일을 보관하거나 프로그람을 설치한 경우에 원래 파일이 사용하던 블로크우에 덧씌워질 가능성이 큰데 이런 경우에는 파일을 복구할수 없다.

삭제할 등록부에로 이동한다.

```
# cd /hdc1/mp3-date
```

삭제할 파일을 확인해본다.

```
kcc@linux:~

# ls
The Palace of Versailles.mp3 Rare Bird.mp3
```

하나의 파일을 삭제해보자. 여기에서는 Rare Bird.mp3파일을 삭제하여보겠다.

```
# rm -rf Rare Bird.mp3
```

debugfs명령을 삭제한 파일이 있는 구획에 적용하여 실행하게 한다.

```
kcc@linux:~

# debugfs /dev/hdc1
debugfs 1.23, 15-Aug-2001 for EXT2 ES 0.5b, 95/08/09
debugfs: lsdel
14 deleted inodes found.
Inode Owner Mode Size Blocks  Time deleted
730193 0   100600 12288  3/   3 Sat Jan 19 21:51:09 2004
730194 0   100700 413    1    1 Sat Jan 19 21:51:09 2004
97472 0    100644  3400515    792/   832 Mon Mar 25 16:18:52 2004
551655 0   100600  12288   3/   3 Mon May 13 13:48:04 2004
503066 0   100644  2408448     589/   589 Tue Jun 4 12:51:15 2004
503077 0   100644  573440141/  141 Tue Jun 4 12:51:47 2004
503091 0   100644  352088      861/  861 Tue Jun 4 12:52:09 2004
503092 0   100644  1894318     464/ 464 Tue Jun 4 12:52:09 2004
503094 0   100644  3440232     841/ 841 Tue Jun 4 12:52:09 2004
503095 0   100644  2870125     702/ 702 Tue Jun 4 12:52:09 2004
503096 0   100644  2975033     728/ 728 Tue Jun 4 12:52:09 2004
503097 0   100644  3609913     883/ 883 Tue Jun 4 12:52:09 2004
503098 0   100644  3662158     896/ 896 Tue Jun 4 12:52:09 2004
649121 500 100644  3125376     765/ 765 Tue Jan 6 18:33:04 2004
```

lsdel 명령으로 삭제된 파일을 볼수 있다.

649121의 inode값을 가진 파일이 2004년 1월 6일 저녁 6시 33분에 삭제된것을 볼수 있다.

```
debugfs: dump <649121> /root/dumptest.dump
```

dump명령으로 649121파일을 다른 파일체계로 복구한다.

```
debugfs: quit
```

이전에 삭제된 파일인 Bird.mp3파일이 dumptest.dump파일로 복구된것을 알수 있다.

```
kcc@linux:~
# ls -al /root/dumptest.dump
-rw-r--r--   1 root    root    3125376  1월 6 18:37 /root/dumptest.dump
```

depmod /sbin/depmod

모듈의존성의 확인

핵심부모듈을 적재하기 위한 의존성관계를 다루는 명령문이다.

형식 : depmod[설정값]모듈명

-a : /etc/modules.conf에 있는 모든 모듈의 의존성을 생성한다.
-d : 오유수정방식
-e : 해결되지 않는 각 모듈의 목록을 보여준다.
-v : 처리된 모듈의 모든 목록을 보여준다.

depmod명령문은 핵심부모듈(kernel module)의 의존성관계를 다루는 명령문이다. 이와 관련된 파일은 /etc/modules.conf(이것은 RedHat 7.0판인 경우이고 다른 배포판이나 판본에서는 /etc/conf.modules파일일수 있다)이다.

사용법은 modprobe명령에 함께 설명되여있다.

【관련명령문】

insmod : 모듈적재명령문
modprobe : insmod 보다 우수한 모듈적재명령문

df

/bin/df

구획의 남은 용량을 보기

현재 mount되어있는 파일체계의 사용정보를 보여주는 명령문이다.

형식 : df[설정값][파일이름...]

-a, -alt : 0블로크의 파일체계를 포함하여 모든 파일체계를 보여준다.

-i, -inodes : inode의 남은 공간, 사용공간, 사용비률 등의 정보를 보여준다.

-k, -kilobytes : 1KB단위로 보여준다.

-P, -portability : POSIX에서 사용되는 형태로 보여준다.

-T, -print-type : 각 파일체계의 형태를 추가하여 각 구획의 정보를 보여준다.

-t **파일체계** : 지정한 파일체계형태만의 정보를 보여준다.

-x **파일체계** : 지정한 파일체계형태만을 제외한 나머지 모든 파일체계정보를 보여준다.

-help : 도움말을 보여준다.

df명령문은 현재 mount되어있는 파일체계의 사용정보를 보여주는 명령문으로서 1024byte의 블로크단위로 보여준다.

```
kcc@linux:~
# df
Filesystem   1K-blocks  Used     Available  Use%  Mounted on
/dev/hda8    8206812    4822520  2967408    62%   /
/dev/hda1    54416      5849     45758      12%   /boot
/dev/hda3    2522076    1966488  427472     83%   /home
none         63020      0        63020      0%    /dev/shm
/dev/hda2    101107     3061     92825      4%    /tmp
/dev/hda7    3028080    388076   2486184    14%   /var
/dev/hdb1    10080488   1379124  8189296    15%   /hdb1
/dev/hdb2    9614148    1441116  7684656    16%   /hdb2
/dev/hdc1    6248020    5075644   854984    86%   /hdc1
```

[주의]

파일체계의 총 용량이 5%이하 남아있는 경우, 일반사용자는 체계에 접속할수 없다. 일정한 인원별로 사용자구획을 나누어 놓는것이 한가지 방법으로 된다.

inode의 사용정보를 보자. 이 inode가 부족하면 사용공간이 남아있어도 더이상 파일을 새롭게 만들수 없으므로 확인해볼 필요가 있다.

```
kcc@linux:~

# df -i
Filesystem    Inodes    IUsed    IFree    IUse%  Mounted on
/dev/hda8     1042432   212661   829771   21%    /
/dev/hda1     14056     38       14018    1%     /boot
/dev/hda3     320640    448      320192   1%     /home
none          15755     1        15754    1%     /dev/shm
/dev/hda2     26208     165      26043    1%     /tmp
/dev/hda7     384768    31028    353740   9%     /var
/dev/hdb1     1281696   5793     1275903  1%     /hdb1
/dev/hdb2     1221600   13251    1208349  2%     /hdb2
/dev/hdc1     794976    5766     789210   1%     /hdc1
```

-h설정값을 사용하면 1024byte단위가 아닌 G(GB), M(MB)의 단위로 쉽게 파일체계의 용량을 볼수 있게 한다.

```
kcc@linux:~

# df -h
Filesystem   Size   Used   Avail  Use%  Mounted on
/dev/hda8    7.8G   4.6G   2.8G   62%   /
/dev/hda1    53M    5.8M   44M    12%   /boot
/dev/hda3    2.4G   1.9G   417M   83%   /home
none         61M    0      61M    0%    /dev/shm
/dev/hda2    99M    3.0M   90M    4%    /tmp
/dev/hda7    2.9M   379M   2.3M   14%   /var
/dev/hdb1    9.6G   1.4G   7.8G   15%   /hdb1
/dev/hdb2    9.2G   1.4G   7.8G   16%   /hdb2
/dev/hdc1    6.0G   4.9G   834M   86%   /hdc1
```

【관련명령문】

du : 디스크사용의 현재 상태를 볼수 있는 명령문

free: 현재의 주기억사용상태를 볼수 있는 명령문

diff

/usr/bin/diff

파일의 다른 점을 비교

비교되는 두 파일에서 다른 부분을 두 파일의 내용과 함께 보여주는 명령문이다.

형식 : diff[설정값]기준파일 비교파일

-lines : 문맥의 행을 보여준다.

-a, -text : 모든 파일을 본문파일로 취급한다.

-b, -ignore-all-space : 중복된 공백과 행의 끝에 있는 공백은 무시한다.

-B, -ignore-blank-lines : 파일에 있는 공백을 무시한다.

-c, -C num, -context[=num] : 문맥상으로 바뀐 행을 n수만큼 보여준다. (기본은 2행이다)

-d, -minimal : 세세하게 바뀐것도 찾도록 한다.

-D name, -ifdef=name : C원천파일내에서 #ifdef로 정의된 name에서도 보여준다.

-e, -ed : ed스크립트를 리용하여 보여준다.

-F regexp, -show-fuction-linux=regexp : 지정한 정규표현식에 맞는 가장 최근의 행부터 보여준다.

-H, -speed-large-files : 큰 파일을 빠르게 처리하기 위해 사용된다.

-help : 간단한 사용법을 보여준다.

-horizon-lines=lines : 공통된 부분의 앞부분이 속한 행의 마지막행과 공통된 끝부분이 속한 행의 첫번째행을 버리지 않는다.

-i, -ignore-case : 대문자와 소문자를 구별하지 않는다.

-I regexp, -ignore-matching-lines=regexp : 지정한 정규표현식에서의 형태로 바뀐 부분은 무시한다.

-l, -paginate : pr을 통해 페지를 할당한다.

-L label, -label=label : 파일이름대신 지정한 label을 사용한다.

-n, -rcs : RCS diff형식으로 보여준다.

-N, -new-file : 존재하지 않는 파일을 빈 파일로 인식한다.

-P, -unidirectional-new-file : 등록부를 비교할 때에 두개의 등록부중 두번째 등록부에만 파일이 존재하는 경우, 첫번째 등록부에는 사실상 파일이 없지만 있는것처럼 처리한다.

-q, -brief : 파일간에 차이가 있는 경우에만 보여준다.
-r, -recusive : 비교대상등록부의 하위등록부나 파일까지 비교한다.
-s, -report-idenfical-files : 두 파일이 같은 경우에 보여준다.
-S file, -starting-file=filename: 등록부비교때에 지정한 파일명을 우선 목록에서 보여준다.
-t, -expand-tabs : 입력파일에서 Tab의 정렬을 지키기 위해 출력시에 Tab의 공간을 확장한다.
-T, -initial-tab : 들여쓰기를 위해 초기의 Tab를 삽입한다.
-u, -U n, -unified[=n]: 문맥상으로 합져진 행을 지정한 n수만큼 보여준다.(기본은 2행이다.)
-v, -version : 판본정보를 보여준다.
-w, -ignore-all-space : 두행의 비교때에 공백을 무시한다.
-W n, -width=n: 한 렬의 최대문자수를 n으로 지정한다. (기본은 130이다.)
-x regexp, -exclude=regexp : 지정한 정규표현식과 같은 파일은 제외한다.
-X filename, -exclude-from=filename : 지정한 파일명과 일치하는 파일과 등록부는 비교하지 않는다.
-y, -side-by-side : 비교결과로서 두개의 렬을 보여준다.

diff명령문은 두 파일의 차이점을 확인하는데 매우 효과적인 명령문이다.
아래의 실례를 통해 보도록 하자. 아래의것은 qmail SMTP원천의 차이점을 보기 위해 매 파일등록부에 있는 chkspawn.c를 분석한것이다.

```
kcc@linux:~

# diff qmail-1.03/chkspawn.c qmail_tuning/chkspawn.c
25, 26c25, 26
>if (auto_spawn > 255) {
>   substdio_puts(subfderr, "Oops, You have set conf-spawn
higher than 255.\n");
—
<if (auto_spawn > 65000) {
<   substdio_puts(subfderr, "Oops. You have set conf-spawn
higher than 65000.\n");
```

[주의]

세 파일을 서로 비교할 때에는 diff3 명령을 리용한다.

25, 26c25, 26은 앞파일의 25, 26행이 뒤파일의 25, 26행에서 변경되였다는 뜻이다. > 표식은 앞의 파일을 나타내며 < 표식은 뒤파일의 내용을 나타낸다.

```
kcc@linux:~

159a162,164
>             delnum+=(unsigned int) ((unsigned int) ch) << 8;
>              messid.len=0; stage=2; break;
>case 2:
```

앞파일의 159번째 행과 비교하여 뒤 파일의 162, 163, 164번째 행에 우의 세행의 내용이 추가된것을 볼수 있다.

이와 같은 방법을 사용하여 어떤 원천프로그람형식의 수정프로그람(Patch)을 구입하였을 때에 이들사이에 어느 부분이 어떻게 달라졌는가를 쉽게 확인할수 있다. 또한 프로그람을 설치할 때마다 어떤 프로그람이 추가되는가를 손쉽게 확인하려는 경우에도 프로그람을 설치한후 find명령을 통해 전체 파일내용을 파일로 보관하고 새로운 프로그람을 설치했을 때마다 diff명령을 사용하여 확인할수도 있다.

참 고

수정파일(Patch File)의 만들기

원천코드를 일부 수정하여 달라진 부분을 손쉽게 적용할수 있도록 만든것을 《수정파일》이라고 부른다. 이것은 앞에서 언급한 림시수정프로그람을 의미한다. 수정파일은 diff를 사용하여 손쉽게 만들수 있다.

Lovesong2.3이라는 프로그람의 원천코드는 lovesong2.3이라는 등록부안에 들어있다. 원래의 원천코드등록부이름을 lovesong2.3-orig로 변경하고 새로 추가한 원천코드등록부이름을 lovesong2.3으로 둔다. 그리고 다음과 같이 실행하면 수정파일을 만들수 있다.

```
# diff -urN song2.3-orig song2.3 > song2.3-new.diff
```

이 diff파일안에는 두 등록부의 원천코드에서 변경된 부분이 비교되여있다.

—로 시작하는 행은 원본파일, +++로 시작하는 행은 수정파일의 내용이다. 그리고 —표시가 붙은 행은 삭제하고 +표시가 붙은 행은 추가하거나 바꾸어 넣는다는 의미로 리해할수 있다.

dirname

/usr/bin/dirname

경로이름보기

전체 경로에서 파일이름을 제외한 등록부의 이름을 보여주는 명령문이다.

형식 : dirname 파일이름

-help : 도움말을 보여준다.

-version : 판본정보를 보여준다.

dirname은 지정한 파일이나 파일등록부의 등록부구조의 마지막준위를 제외한 파일등록부의 경로를 보여준다. 만일 전체 절대경로를 지정하지 않으면 현재 파일등록부인 《.》을 보여주게 된다.

```
kcc@linux:~
# dirname Work/
.
# dirname /root/Work/
/root
```

참 고

절대경로와 상대경로

콤퓨터조작체계에서 path란 파일체계를 통해 특정한 파일에 이르는 경로를 말한다. 경로이름이란 경로를 자세히 렬거한것이다. 매 조작체계들에서는 경로이름을 지정하기 위한 고유한 형식을 가지고있다. UNIX체계에서 경로의 형식은 다음과 같다.

```
/directory/subdirectory/filename
```

절대경로이름이란 경로의 처음부터 끝까지를 완전히 서술한 경로이름을 말한다.

상대경로이름은 현재 설정해놓은 등록부로부터 시작하여 상대적인 경로를 지정한다.

dmesg

/bin/dmesg

기동리력의 보기

핵심부순환완충기(kernel loop buffer)를 보고 조종하는 명령문이다.

형식 : dmesg[설정값]

c : 통보문내용을 보여주고 순환완충기를 모두 비운다.
-n level : 지정한 lebel 의 값에 한해서만 완충기의 내용을 보여준다.

dmegs는 체계를 기동할 때에 적재되는 정보들의 각종 통보문을 가지고있으며 이것을 다시 볼 때에 리용한다. 이 정보에는 Linux핵심부판본, gcc판본, CPU정보 등 기동시의 체계정보를 재확인할수 있게 한다.

```
kcc@linux:~

# dmesg | more
Linux version 2.4.13-1hl (root@lhcha.kcc.com)  (gcc version 2.96
20000731 (Red Hat Linux 7.1 2.96-98))
#1 2001.11. 04 (일) 04:04:58 KST
BIOS-provided physical RAM map:
BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
BIOS-e820: 0000000000100000 - 0000000000800000 (usable)
BIOS-e820: 00000000ffff0000 - 0000000100000000 (reserved)
Scanning bios EBDA for MXT signature
user-defined physical RAM map:
user: 0000000000000000 - 000000000009fc00 (usable)
user: 000000000009fc00 - 00000000000a0000 (reserved)
user: 00000000000f0000 - 0000000000100000 (reserved)
user: 0000000000100000 - 0000000008000000 (usable)
user: 00000000ffff0000 - 0000000100000000 (reserved)
On node 0 totalpages: 32768
zone(0): 4096 pages.
zone(1): 28672 pages.
```

```
zone(2): 0 pages.
Kernel command line: ro root=/dev/hda8 mem=131072K
Initializing CPU#0
Detected 501.145 MHz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 999.42 BogoMIPS
Memory: 124908k/131072k available (1922k kernel code, 4876k
reserved, 97k data, 23
2k init, 0k hignmem)
Dentry-cache hash table entries: 16384 (order: 5, 131072 bytes)
Inode-cache hash table entries: 8192 (order: 4, 65536 bytes)
Mount-cache hash table entries: 2048 (order: 2, 16384 bytes)
Buffer-cache hash table entries: 8192 (order: 3, 32768 bytes)
Pages-cache hash table entries: 32768 (order: 5, 131072 bytes)
CPU: Before vendor init, caps: 0387f9ff 00000000 00000000, vendor= 0
CPU: L1 I cache: 16K, L1 D cache: 16K
CPU: L2 cache: 512K
CPU: After vendor init, caps: 0387f9ff 00000000 00000000 00000000
CPU serial number disabled.
Intel machine check architecture supported.
Intel machine check reporting enabled on CPU#0.
CPU:      After generic, caps: 0383f9ff 00000000 00000000 00000000
CPU:             Common caps: 038f9ff 00000000 00000000 00000000
CPU: Intel Pentium Ⅲ (Katmai) stepping 03
Enabling fast FPU save and restore... done.
Enabling unmasked SIMD FPU exception support... done.
Checking 'hlt' instruction... OK.
POSIX conformance testing by UNIFIX
Mtrr: v1.40 (20010327) Richard Gooch (rgooch@atnf.csiro.au)
mtrr: detected mtrr bype: Intel
PCI: PCI BIOS revision 2.10 entry at 0xfb270, last bus=1
PCI: Using configuration type 1
PCI: Probing PCI hardware
Unknown bridge resource 0: assuming transparent
PCI: Using IRQ router VIA [1106/0596] at 00:07.0
Activating ISA DMA hang workarounds.
isapnp: Scanning for PnP cards...
isapnp: No Plug & Play device found
—more—
```

우의 정보로부터 이 체계는 Linux kernel 2.4.13을 사용하고있으며 gcc 2.96, 주기억기가 128M정도, CPU는 Intel PentiumⅢ(Katmai), ISA확장홈과 관련된 기판은 발견되지 않는것 등을 알수 있다.

참 고

핵심부판본(kernel version)의 의미

먼저 현재 최신의 핵심부판본을 확인하여보자

```
# finger @finger.kernel.org
```

우와 같이 하면 최신의 안정판본과 개발판본을 확인할수 있다. 실례로 《2.4.19 version》 다음의 세 부분으로 나누어 의미를 나타낸다.

2 : 몇번의 림시수정프로그람(Patch)이 제공되였는가를 나타낸다.
4 : 안정판본인가 혹은 개발판본인가를 나타낸다(홀수인 경우에는 개발판본을 의미하며 짝수인 경우에는 안정판본을 의미한다).
19 : 획기적인 변화가 있을 때 바뀐다.

우의 의미대로 보면 kernel 2.4.19의 경우, 2.4세대의 안정판본으로서 19번의 공식적인 림시수정프로그람(Patch)이 추가되였다는것을 알수 있다.

개발판본은 충분한 림시수정프로그람의 제공과 시험을 통해 안정성이 검증되면 짝수판본, 즉 안정판본으로 바뀐다. 핵심부개발자가 아닌 이상은 안정판본을 사용하는것이 일반적이다.

dnsdomainname /bin/dnsdomainname

DNS령역확인

체계의 DNS봉사기이름과 정보를 볼수 있는 명령문이다.

형식 : dnsdomainname[-v]

dnsdomainname명령문은 TCP/IP명령문으로서 체계의 DNS봉사기에 대한 정보를 보여준다.

이 체계의 DNS봉사기를 확인해보자.

```
kcc@linux:~
# dnsdomainname
linuxroot.co.kp
```

-v 설정값을 입력하면 보다 상세한 정보를 보여준다.

```
kcc@linux:~
# dnsdomainname -v
gethostname()=ns.kccredstar.co.kp'
Resolving ns.kccredstar.co.kp' ...
Result: h_name=ns.kccredstar.co.kp'
Result: h_addr_list=61.40.233.122'
kccredstar.co.kp
```

【관련명령문】

hostname : 체계의 호스트이름을 설정하거나 보여주는 명령문
domainname : 체계의 NIS/YP령역이름을 설정하거나 보여주는 명령문
nisdomainname : 체계의 NIS/YP령역이름을 설정하거나 보여주는 명령문
ypdomainname : 체계의 NIS/YP령역이름을 설정하거나 보여주는 명령문

domainname

/bin/domainname

NIS령역의 보기

체계의 DNS는 물론 NIS/YP령역이름을 설정하거나 보여주는 명령문이다.

형식 : domainname[설정값][이름]

-s, -short : 간략화한 호스트이름을 보여준다.
-a, -alias : alias이름을 보여준다.
-I, -ip-address : 호스트이름의 IP주소를 보여준다.
-f, -fqdn, -long : full호스트이름을 보여준다.
-d, -domain : DNS령역이름을 보여준다.
-y, -yp, -nis: NIS/YP령역이름을 보여준다.
-F, -file: file로 지정된 파일로부터 읽어 호스트이름이나 NIS령역이름을 보여준다.

domainname명령문은 NIS/YP명령문으로 현재의 NIS령역이나 설정된 내용을 보여주는 명령문이다. 령역이름을 변경하여본다.

```
kcc@linux:~
# domainname
linuxroot.co.kp
# domainname -s
ns
# domainname -i
61.40.233.122
# domainname -f
ns.linuxroot.co.kp
```

```
kcc@linux:~
# domainname test.co.kp
# domainname
test.co.kp
```

【관련명령문】

hostname : 체계의 호스트이름을 설정하거나 보여주는 명령문
dnsdomainname : 체계의 DNS봉사기이름과 정보를 보여주는 명령문
nisdomainname : 체계의 NIS/YP령역이름을 설정하거나 보여주는 명령문
ypdomainname : 체계의 NIS/YP령역이름을 설정하거나 보여주는 명령문

참 고

NIS/YP

NIS는 Sun회사가 개발한 망봉사로서 Netrwork Infor-mation service(망정보봉사)의 략자이다. 원래는 YP(Yellow Pages)라는 이름으로 불리웠는데 Yellow Pages라는 이름이 이미 등록되여있어 허가없이는 사용할수 없는것으로 하여 이후에 NIS라는 이름으로 등록하였다. NIS는 Sun Mocrosystems, Inc.의 등록상표이다.

NIS는 망에서 다른 콤퓨터에 정보를 전달하는 방식으로 망우에 있는 모든 체계에 봉사기의 /etc/passwd에 보관된 망가입자이름과 암호, home등록부정보와 /etc/group에 보관된 집단정보를 제공하는것이다. NIS호스트봉사기는 정보표를 소유하며 그것들을 NIS대응(map)파일들로 변환한다. 이 대응파일들이 망을 통해 제공됨으로써 NIS의뢰기콤퓨터들은 우와 같은 여러 정보들을 얻을수 있는것이다. NIS를 리용하면 NIS자료기지에 등록된 사용자들은 이 망에서 NIS의뢰기가 설치되여있는 어떤 체계에도 망접속할수 있으며 암호를 한번만 바꾸면 그 NIS령역에 들어있는 모든 콤퓨터의 망접속정보를 갱신할수 있다.

NIS는 안전한것이 못된다. 원래부터 안전측면보다 사용편리를 중시하여 만든것이였으므로 NIS의 사용에서는 보안에 세심한 주의를 돌려야 한다.

봉사기의 NIS령역이름을 알아내려고 하는 부정사용자의 경우, 여러가지 방법을 사용하여 passwd파일복사본을 얻거나 사용자암호를 알아내기 위해 다양한 암호침입도구를 사용할수도 있다. 또한 NIS를 속여넘겨 여러 형태의 불법행위들을 저지를수 있다. 이러한 점을 조금이라도 방지하기 위해서는 NIS+를 사용할수 있다.

NIS+에 대해서는 해당한 관련문서(Sun회사나 다른 여러 싸이트들에서 제공)를 참고하기 바란다.

dosfsck

/sbin/dosfsck

MS-DOS파일체계의 복구

MS-DOS파일체계를 검사하고 복구하는 명령문이다.

형식 : dosfsck[설정값...]장치명

-a : 파일체계를 자동적으로 복구한다.
-d file : 파일 할당표에서 지정한 파일을 버린다.
-f : 손실된 사슬(chain)을 파일로 복구한다.
-l : 경로이름을 보여준다.
-r : 파일체계복구때에 일일이 물어본다.
-t : 검사를 진행한다.
-u file : 지정한 파일을 삭제하지 않는다.
-v : 보다 상세하게 보여준다.
-V : 확인절차를 생략한다.
-w : 변경된 내용을 즉시 디스크에 쓰기한다.

dosfsck명령문은 체계관리명령문으로서 fsck와 비슷하다. 그러나 이것은 MS-DOS파일체계에만 적용할수 있다.

Linux는 MS-DOS, Sun 등의 다양한 환경에서 파일체계의 읽기 및 쓰기를 할수 있도록 지원한다. 이것은 다양한 조작체계들에서 사용하는 파일체계를 지원하여 사용자들의 불편을 최소화할것을 지향하고있다. 그러나 파일체계를 검사하고 복구하는것은 매 고유한 파일체계별로 특징이 있기때문에 서로 다른 명령이 필요하다.

Linux에서는 많이 사용하는 다른 조작체계의 파일체계를 검사하고 복구하는 명령도 활용 되고있는데 그중 대표적인것으로는 FAT파일체계에 대한것과 MINIX파일체계에 대한것을 들수 있다.

du

/usr/bin/du

디스크사용정형을 보기

현재의 디스크사용상태를 보여주는 명령문이다.

형식 : du[설정값]

a, -all : 현재 등록부아래의 모든 파일과 등록부의 사용정보를 보여준다.
-B, -block-size : 지정한 size를 블로크크기로 사용한다.
-b, -bytes : 바이트크기로 보여준다.
-c, -total : 모든 파일의 디스크사용정보와 합계를 보여준다. 이것은 그 해당경로가 얼마만큼 디스크공간을 사용하고있는가를 확인하는데 사용한다.
-D, -dereference-args : 계산되는 파일이나 경로가 기호련결파일인 경우, 그 원본의 값을 보여준다.
-h, -human-reabable : 파일용량을 사람이 읽기 쉬운 형태로 보여준다.
-H, -si : -h와 비슷한데 1024단위의 비률로 보여준다.
-k : Kbyte단위의 형태로 보여준다.
-l, -count-links : 계산되는 파일중에 하드련결되여있는 파일이 있는 경우, 그것을 그대로 계산한다.
-L, -derefernce : 모든 기호련결을 따른다.
-S, -separate-dirs : 등록부의 총 사용량을 보여줄 때에 하위등록부의 사용량은 제외한다.
-s, -summarize : 간단히 총사용량만을 보여준다.
-x, -one-file-system : 현재 파일체계의 파일사용량만을 보여준다.
-X file, -exclude-from=file : 지정한 파일과 일치하는것은 제외한다.
-help : 도움말을 보여준다.
-version : 판본정보를 보여준다.

현재 등록부에서 du명령은 사용량을 보여준다(기본단위는 Mbyte형태이다).
현재 경로의 모든 등록부의 크기를 나타내는데 -a설정값을 사용하면 모든 파일을 보여주며 지정한 파일이나 등록부를 요약하여 나타낼수도 있다.

-s설정값을 사용하면 아래와 같이 지정한 파일이나 등록부의 총 사용량만을 보여준다.

```
kcc@linux:~

# du -s /home/*
1666108 /home/admin
4       /home/ftp
16      /home/list+found
3200     /home/oracle
128     /home/test
```

du명령의 -c설정값은 현재 등록부의 총 사용량을 보여주며 h는 알아보기 쉽게 적당한 단위별로 보여준다.

```
kcc@linux:~

# du -ch
8.0K    ./.ami
8.0K    ./.kde/Autostart
8.0K    ./.kde/share/apps/konqueror/profiles
16K     ./.kde/share/apps/konqueror/dirtree/remode/ttp
24K     ./.kde/share/apps/konqueror/dirtree/remode/web
52K     ./.kde/share/apps/konqueror/dirtree/remode
--------중간 생략------------------------------
1.4M    ./KccOffice2.0-eval/rc/kor
1.4M    ./KccOffice2.0-eval/rc
240M    ./kccOffice2.0-eval
12K     ./.w3m
240K    ./expolit
736M    .
736M    합계
```

dumpe2fs

/sbin/dumpe2fs

파일체계정보의 보기

ext2 파일 체계정보를 보여주는 명령문

형식 : dumpe2fs[장치명]

dumpe2fs 명령문은 장치명에 존재하는 파일체계의 관리정보블로크(Super Block)와 블로크집단의 정보를 보여준다.

```
kcc@linux:~

# dumpe2fs/dev/hda8
dumpe2fs 1.23, 15-Feb-2004 for EXT2 FS 0.5b, 95/08/09
Filesystem volume name:                    /
Last mounted on:                           <not available>
Filesystem UUID: c332b42a-f077-11d5-9480-e4137dff02da6
Filesystem magic number:                   0xEF53
Filesystem revision #:                     1(dynamic)
Filesystem features:                       filetype sparse_super
Filesystem state:                          not clean
Errors behavior:Continue
Filesystem OS type:                        Linux
Inode count:    1042432
Block count:    2084425
Reserved block count:   104221
Free blocks:     745992
Free inodes:     830221
First block:     0
Block size:      4096
Fragment size:   4096
Blocks per group:                          32768
Fragments per group:                       32768
Inodes per group:                          16288
Inode blocks per group:                    509
Last write time: Tue Apr 2 00:07:03 2004
Mount count:     1
Maximum mount count:                       35
```

```
Last checked: Mon Apr 24 15:20:06  2004
Check interal: 15552000 (6 months)
Next check after: Sat Jun 21 15:20:06 2004
Reserved blocks uid:                          0(user root)
Reserved blocks gid:                          0 (group root)
First inode:     11
Inode size:      128
---------------------아래 생략------------------
```

참 고

Linux파일체계

Linux파일체계는 5개의 주요부분으로 이루어져 있다. 파일체계의 앞부분에는 기동블로크(Boot Block)가 보관된다. 기동블로크를 제외한 나머지 부분은 실린더집단(Sylinder Group)이라고 하며 여러 실린더집단으로 나누어 관리한다. 하나의 실린더집단은 관리정보블로크(Super Block), 실린더집단블로크(Sylinder Group Block), inode표(i-node Block), 자료블로크(Data Block) 등으로 이루어져 있다.

(1) **기동블로크:** 기동에 필요한 파일들이 존재하며 뿌리령역외에는 해당되지 않는다. 일반 사용자에게는 해당되지 않는 블로크이다.

(2) **관리정보블로크:** 파일체계의 크기, i-node표의 크기, free블로크목록 등 파일체계를 관리하는데 필수적인 정보가 보관된다. 이 부분에 오유가 생기면 이 실린더를 사용할수 없게 되므로 일반적으로 관리정보블로크는 여러 블로크에 걸쳐 예비보관(Backup)을 만든다.

(3) **실린더집단블로크:** 실린더집단의 유효블로크들의 비트정보나 통계정보를 기록한다.

(4) **inode표:** 파일에 대한 중요한 정보를 기록한다. 파일의 크기, 자료위치(디스크주소), 파일형식, 사용허가권(License), 생성날자 등 핵심부(Kernel)의 파일관리에 있어서 핵심으로 되는 모든 자료를 기록한다.

(5) **자료블로크:** 실지 자료가 보관되는 공간이다.

dumpkeys

/usr/bin/dumpkeys

건반코드값보기

건반의 입력을 코드값으로 변환하는 표에 관한 정보를 보여주는 명령문이다.

형식 : dumpkeys [설정값]

-i, -shof-info : 건반구동프로그람에 관한 정보를 보여준다.

-l, -long-info : -i 설정값에 추가적으로 핵심부(kernel)에서 지원가능한 건의 수와 상세정보를 보여준다.

-n, -numertc : 16진수표시법으로 활성화된 코드값을 보여준다.

-f, -full-table : 간략한 형식이 아닌 완전표(full table)를 보여준다.

-l, -separate-lines : 매 modifier/keycode를 한행에 하나씩 보여주며 plain을 변경되지 않는 keycode의 접두사로 한다.

-s, -shape=num : 표형태를 결정하는 num값을 리용하여 정보를 보여준다. 매 num 값은 다음과 같다.

0	기본값
1	--full-table 과 같다.
2	--separate-lines 와 같다.
3	keycode 를 첫번째 행에 두고 modifier/keycode 부분을 한행씩 보여준다.

-c, -charset=(charset-name) : 지정한 문자모임을 활성화코드로 해석한다. 아래와 같은 문자모임(Character Set)들을 사용할수 있다.

iso-8859	1iso-8859-2	iso-8859-3	iso-8859-4	iso-8859-5
iso-8859-7	iso-8859-8	iso-8859-9	iso-8859-14	iso-8859-15
mazovia	koi8-r	koi8-u	iso-10646-18	

-h, -help : 도움말을 보여준다.

-v, -version : 판본정보를 보여준다.

dumpkeys명령문은 많이 사용하지 않는 명령문이지만 keycode값과 정보를 확인하거거나 보는데 유용하다.

```
kcc@linux:~

# dumpkeys l more
keymaps 0-2,4-5,8,12
keycode       1=Escape          Escape
              alt      keycode    1=Meta_Escape
keycode       2=one             exclam
              nul      keycode    2=Meta_one
keycode       3=two             at   at         nul
              nul               Meta_Escape
keycode       4=three           numbersign
              control  keycode    4=Escape
              alt      keycode    4=Meta_three
keycode       5=four            dollar  dollar     Control_back
slash
              alt      keycode    5=Meta_four
keycode       6=five   percent
              control  keycode    6=Control_bracketright
              alt      keycode    6=Meta_five
keycode       7=six             asciicircum
              control  keycode    7=Control_asciicircum
              alt      keycode    7=Meta_six
keycode       8=seven           ampersand   braceleft   control_unde
rscore
              alt      keycode    8=Meta_seven
—More—
```

E

e2fsck /sbin/e2fsck

Linux파일체계검사

ext2파일체계를 검사하는 명령문이다.

형식 : e2fsck[설정값]장치이름, fsck.ext2[설정값]장치이름

-b superblock: 기본 superblock를 사용하지 않고 지정한 superblock를 불러온다.

-B blocksize: 관리정보블로크를 찾기 위해 지정한 블로크크기를 사용한다.

-c: 불량블로크를 검사한다.

-f: 파일체계의 이상유무에 관계없이 파일체계를 검사한다.

-l bad_blocks_file: 지정한 불량블로크파일을 불량블로크목록에 추가한다.

-L bad_block_file: 지정한 불량블로크파일을 불량블로크목록에 설정한다.

-n: 파일체계를 읽기전용으로 열고 질문에 대해 no로 한다.

-p: 사용자의견을 묻지 않고 자동적으로 파일체계를 검사한다.

-n: 파일체계를 읽기전용으로 열고 질문에 대해 no로 한다.

-v: 상세한 정보를 보여준다.

-y: 파일체계를 읽기전용으로 열고 질문에 대해 yes로 한다.

e2fsck명령문은 fack명령문에 -t ext2설정값을 추가하여준것과 같은 명령문이다. fsck명령이나 e2fsck명령문은 파일체계를 직접 검사하는 명령이므로 체계에 련결되여있는 상태보다는 mount를 해제한 상태에서 파일체계를 검사할것을 권고한다. 물론 의도적으로 파일체계를 검사하는 경우도 있다.

그러나 불의적인 상태로 인하여 체계에 이상이 발생한 경우, 기동시에 대부분의 체계에서는 e2fsck가 자동적으로 파일체계의 검사를 진행한다.

문제가 심각한 경우에는 파일체계의 수리중에 다음과 같은 통보문을 보여주어 중단하는 경우도 있다.

```
kcc@linux:~

***An error occurred during the file system check.
***Dropping you to a shell; the system will reboot
***when you leave the shell.
Give root password for maintence.
(or type Control-D for normal startup): => root 암호입력
(Repair filesystem) #
```

우의 통보문과 같이 뿌리 password를 입력하면 입력독촉(prompt)이 화면상에 표시되는것을 볼수 있다. 만일 그대로 다시 기동하려면 Control-D 건을 입력한다. 이러한 경우 대부분은 뿌리 password를 입력하여 입력독촉상태로 하여준 다음 파일체계를 검사하여 해결한다. 이 상태에서 파일체계를 검사하면 된다.

```
(Repair file system)# fsck/dev/hda1
```

여기에서 /dev/hda1는 Linux가 설치되여있는 구획을 말하는데 각자의 환경에 따라 달라질수 있다. 만일 mount된 /dev/hdc1 파일체계를 확인하는 경우, 먼저 /hdc1의 mount를 해제한 다음 실행하자.

```
kcc@linux:~
# umount/hdc1
# e2fsck /dev/hdc1
e2fasck 1.23, 15-Aug-2001 for EXT2 FS 0.5b, 95/08/09
/hdc was not cleanly unmounted, check forced.
Pass 1:Checking inodes, blocks, and sizes
Pass 2:Checking directory structure
Pass 3:Checking directory connectivity
Pass 4:Checking reference counts
Pass 5:Checking group summary information
/hdc:5766/794976 files (1.9% non-contiguous), 1293868/1586962
blocks
```

[주의]

만일 구획에 대해 제대로 알지 못하는 경우에 fdisk -l명령으로 확인해볼수 있다.

【관련명령문】

fsck : 파일체계를 검사하는 명령문

echo

내부명령문

환경변수출력

체계의 환경변수 또는 입력내용을 화면에 출력한다.

형식 : echo[-ne][문자렬],echo{--help,--version}

-n : 마지막에 개행 (newline: 화면상에서 유표를 한행 아래로 내리우도록 하는 코드로서 ASCII문자표에서 10진수로 〈10〉에 해당한 값을 가진다)문자를 출력하지 않는다.

-e : 문자렬에서 역사선(\)문자와 조합되는 절환문자(escape sequence: 인쇄기나 화면의 속성을 조종하기 위해 장치에 보내는 특수문자렬을 의미하는데 그 중에서도 특히 escspe문자로 시작되는 기계어명령을 말한다)을 인용부호("")로 묶어 문자를 인식하도록 한다. 아래에 준것이 사용가능한것들이다.

\a	경고음(bell)
\b	뒤로 지우기(backspace)
\c	마지막 개행문자를 사용하지 않는다.
\f	form feed(인쇄기에서 종이이송)
\n	개행문자
\r	복귀(carriage retrum: 유표를 그 행의 맨 앞으로 옮기는데 사용되는 특수한 코드이다. 아스키문자표에서 CR는 10진수로 "13"에 해당되는 값이다.)
\t	수평Tab
\v	수직Tab
\\	역사선(backslash)
\nnn	ASCll코드가 nnn(8진수)인 문자

-help : 표준출력으로 사용법을 출력하고 정상적으로 완료한다.

-version : 표준출력으로 판본정보를 출력하고 정상적으로 완료한다.

[주의]

여기에서 사용하는 escape문자는 C언어의 printf문에서 사용하는 설정값과 같다.

echo명령문은 뒤에 적어준 문자렬을 출력하는 명령문으로서 문자렬을 출력한 후에는 개행문자를 덧붙여 출력한다. 또한 쉘에서 자체적으로 포함하고 있는 내부환경변수를 볼 때에 주로 사용한다.

먼저 내부에서 사용하는 환경변수를 보자.

```
kcc@linux:~

# set | more
BASH=bin/bash
BASH_ENV=/root/.bashrc
BASH_VERSINFO=([0]="2" [1]= "05" [2]= "10"
[3]= "1" [4]= "release" [5]= "i386-
redhat-linux-gnu")
BASH_VERSION=$'2.05.10(1)-release'
COLORS/etc/DIR_COLORS
COLUMNS=82
DIRSTACK=()
EUID=0
GROUPS=()
HISTFILE=/root/.bash_history
HISTFILESIZE=1000
HISTSIZE=1000
HOME=/root
HOSTNAME=ns.linuxroot.co.kp
HOSTTYPE=i386
IFS=$ '\t\n'
INPUTRC=/etc/inputrc
JLESSCHARSET=ko
KDEDIR=/usr
LAMHELPFILE=/etc/lam/lam-helpfile
LANG=ch_KPeucKP
LC_ALL=ch_KreucKP
LD_LIBRARY_PATH=/usr/lib/qt/lib:
LESSOPEN=$ '1/usr/bin/lesspipe.sh %s'
LINES=30
LOGNAME=root
```

echo명령으로 HOME환경변수를 출력해보자. 환경변수를 출력하려면 $문자를 붙여야 한다.

```
kcc@linux:~
# echo $HOME
/root
```

환경변수를 보지 않고 자체문자렬을 출력하려면 다음과 같이 한다.

```
kcc@linux:~
# echo HOME
HOME
```

echo명령문에서 보여지는 개행문자를 삭제해보자.

```
kcc@linux:~
# echo -n HOME
HOME#
```

echo -e를 사용하여 escape문자를 인식할수도 있다. 이때에는 체계에서 경고음을 들을수 있다.

```
# echo-e "\a"
```

매개의 \t\n\r를 사용할수도 있다.

```
kcc@linux:~
# echo-e "test\ttest\ntest\rtest"
test	test
test
# echo "test\ttest\ntest\rtest"
test\ttest\ntest\rtest
```

eject

/usr/bin/eject

CD-ROM자동반출

기록매체의 mount상태를 해제한 다음 반출되게 하는 명령문이다. Linux설치가 끝난후 CD가 자동적으로 튀여나오는것도 바로 이 명령을 사용한것이다.

형식 : eject[설정값][장치]

-h, -help : 간단한 사용법을 보여준다.

-v, -verbose : 지령이 실행되는동안 보다 자세한 설명을 보여준다.

-d, -defeult : eject가 실행되는 기본장치의 이름을 보여준다.

-a, -auto on|1|off|0 : auto-eject방식을 on\off 설정한다.

-c, -changerslot<slot> : ATAPI/IDE CD-ROM변환기에서 CD확장홈을 선택할수 있다. 이것은 Linux Kernel 2.0 이상에서 동작한다. CD-ROM구동기가 동작하는 동안(자료CD가 mount되였거나 음악CD가 재생되고있는 중)에는 쓰일수 없다 (참고: 첫번째 확장홈은 1이 아니라 0이다).

-t, -trayclose : CD-ROM받침틀(CD-ROM Tray)을 닫는 명령문인데 모든 장치에 지원되는것은 아니다.

-x, -cdspeed <speed> : CD-ROM의 속도를 선택하는 지령이다.

speed는 만일 8배속인 경우에 8을 입력하는데 모든 장치에 지원되는것은 아니다.

-n, -noop : 실제 실행되는것이 아니라 반출(eject)되는 장치를 보여준다.

-r, -cdrom : CD-ROM을 반출한다.

-s,--scsi : SCSI장치를 반출한다.

-f, -floppy : 플로피디스크를 반출한다.

-p, —proc : /etc/mtab대신 /proc/mounts파일을 리용한다.

-v, —version : 판본정보를 보여준다.

eject명령문은 **CD-ROM**, 플로피디스크, 테프기억매체, JAZ, ZIP디스크등의 기억매체를 명령문조종으로 제거(반출)할수 있게 한다.

이것은 다중 CD-ROM변환기나 자동반출(auto-eject)을 지원하는 장치들을 조종할수도 있으며 일부 CD-ROM구동기의 디스크받침틀을 닫을수도 있게 한다.

eject명령문을 주면 기본장치인 CD-ROM이 구동기로부터 반출된다.

```
# eject
```

-t 설정값에 의해 CD-ROM받침틀을 한번 넣어보자.

```
# eject-t
```

-d 설정값을 사용하면 반출(eject)이 실행되는 기본장치이름을 볼수 있다.

```
kcc@linux:~
# eject -d
eject:default device: cdrom
```

[주의]

Linux설치가 끝날 때 화면에 현시되는 내용중의 마지막 부분에서 eject가 실행되는것을 확인할수 있다.

참 고

SCSI(small computer system interface)

SCSI란 콤퓨터에서 주변장치를 접속하기 위한 직렬방식의 표준적인 장치대면부로서 입출력모선을 접속하는데 필요한 기계적, 전기적인 요구사항과 모든 주변장치를 중심으로 명령문집합에 대한 규격을 말하며 보통 《스카시》라고 읽는다.

SCSI는 주변장치의 번호만 지정해주면 자료의 충돌문제를 관심하지 않고도 주변장치를 조종할수 있기때문에 널리 사용되고있다.

가격이 비싸고 주변장치가 사용하는 규약이 조금이라도 다르면 사용할수 없는것이 결함이지만 계속 갱신되고있다.

env

내부명령문

환경변수보기

환경변수를 보고 기본 환경변수값을 바꾸는 명령문이다.

형식 : env[설정값][변수=값…][명령[인수…]]

-, -i, -ignore-environment : 현재 환경을 무시하고 지정한 《변수=값》을 따른다.

-u, -unset name : 지정한 변수를 제거한다.

-help : 사용법을 보여준다.

-version : 판본정보를 보여준다.

env명령은 사용자의 현재 환경을 보여주거나 환경변수를 새로운 값으로 설정하고 변경된 환경을 보여주는 명령문이다.

명령을 주면 이 명령은 변경된 환경에서 실행된다.

```
kcc@linux:~

# env
PWD=/root
HOSTNAME=ns.linuxroot.co.kp
LD_LIBRARY_PATH=/usr/lib/qt/lib;
PVM_RSH=/usr/bin/rsh
QTDIR=/usr/lib/qt
LESSOPEN=1/usr/bin/lesspipe.sh %s
MANPATH=/usr/lib/qt/man:
-----------중간 생략---------
SHELL=/bin/bash
USERNAME=root
HOSTTYPE=i386
QT_XFT=no
OSTYPE=linux-gnu
HISTSIZE=1000
LAMHELPFILE=/etc/lam-helpfile
PVM_ROOT=/usr/share/pvm3
```

```
TERM=vt100
HOME=/root
PATH=/usr/lib/qt/bin:/usr/kerberos/sbin:/usr/kerb
eros/bin:/bin:/sbin:/usr/bin:/usr/sbin:usr/local/bin
:/usr/local/sbin:/usr/bin/X11:/usr/X11R6/bin:/root/
bin
JLESSCHARSET=ko
_=/usr/bin/env
OLDPWD=/mnt/cdrom/kcccom/RPMS
```

【관련명령문】

export : 환경변수를 확인하고 수정하는 명령문

ex /bin/ex

행편집기

행단위의 대화식본문편집기이다.

ex는 행단위의 대화식본문편집기로서 지금은 사용되지 않지만 vi편집기에 내장되여 기존 ex편집기의 여러 기능들이 vi에서도 사용되고있다. 이로부터 ex편집기의 기능은 vi에서 설명한다.

참 고

ed에서 vi까지

ed	UNIX의 기본적인 행단위편집기
ex	발전된 형태의 ed(Extended ed)
vi	화면편집기로서 ex명령을 활용한다.
vi클론	발전된 형태의 vi로서 대표적인것은 vim, elvis 이다.

핵심부에 자기이름의 넣기

uname -a로 배포판들의 핵심부판본을 확인해보면 2.4.13-1 혹은 2.4.13-1h1과 같이 핵심부판본뒤의 추가사항을 볼수 있다. 핵심부를 번역할 때 핵심부정보에 자신의 머리글자(Initial)나 원하는 단어를 넣을수 있다.

핵심부원천에서 Makefile이라는 파일을 열어보면 《EXTRAVERSION=》라는 부분이 있다. 이 뒤에 원하는 내용을 적으면 핵심부정보에 입력한 내용을 추가할수 있다. 레를 들어 다음과 같이 입력하고 보관해보자.

```
EXTRAVERSION=-linuxguru
```

핵심부번역후 uname -a명령을 사용하여 핵심부판본을 확인해보면 2.419-linuxguru와 같이 표시된다.

expand

/usr/bin/expand

문서변환

tab를 빈칸으로 만들어주는 명령문이다.

형식 : expand[설정값][파일이름…]

-t, -tabs=NUMBER : tab간격을 조절한다. 기본은 8칸으로 되어있으나 지정한 수자에 따라 tab칸을 조정가능하게 한다.
-help : 사용법을 보여준다.
-version : 판본정보를 보여준다.

expand는 파일내의 tab를 적당한 공백의 수로 변환하는 역할을 한다. 실례를 통해 간단히 보기로 하자.

아래의 내용을 가진 test파일이 있다. tab로 구분되여있는것은 3번째행의 1234와 56사이, 4번째행의 123과 456사이,456과 789사이 등이다.

```
kcc@linux:~
# cat test
123486789
12 34 56789
1234     56  789
123      456     789
```

이것을 expand명령을 사용하여 tab간격을 조절해보자. tab간격을 3칸으로 지정하였다.

```
kcc@linux:~
# expand -t 3 test
123486789
12 34 56789
1234     56  789
123    456    789
123456789
```

```
12 34 56789
1234        56 789
123         456       789
```

참 고

아래준위등록부안의 파일내용을 모두 수정하기

여러개의 파일안에 있는 《개성시》라는 글자를 《남포시》라고 수정할 때에 아래준위등록부에 있는 파일들까지 모두 수정할수 있으면 매우 편리하다. 그러면 먼저 《개성시》라는 문자가 들어있는가를 아래준위등록부의 파일까지 모두 검색해보자.

```
grep 개성시 * -rls
```

특정한 문자렬을 다른 문자렬로 변경할 때에는 vi를 사용하여 다음과 같이 할수 있다. -c 설정값을 사용하면 파일을 열면서 동시에 지시한 vi명령을 수행한다.

```
vim-c : %s/개성시/남포시/g
```

이 두가지를 한번에 련결하여보자. 다음과 같이 for 문을 사용하여 한행에 입력하는것만으로도 모든 작업을 수행할수 있다.

```
for I in 'grep 개성시' -rls ; do(vim-c:%s/개성시/남포시/g-c:wq $i);done
```

expr /usr/bin/expr

표현식평가

지정한 인수를 표현식으로 인식하는 명령문이다.

쉘스크립트를 만들 때에 참고할만한 명령문이다. 자세한 사용법은 쉘스크립트에서 설명한다.

[주의]

하나하나의 항목을 띄여쓰기할 때에만 요구하는 결과를 얻을수 있다. 《1 + 2》가 아니라 《1+2》로 기입한 경우에 문자렬로 인식해버린다.

실례로 이 명령은 주어진 인수를 계산가능한 표현식으로 인식하여 계산이 가능하게 하는 명령이다. 간단한 례로 쉘우에서 더하기연산을 해보자.

```
kcc@linux:~

# expr 1 + 2
3
```

일반적으로는 문자렬로 인식해버릴 가능성이 있는 내용을 계산가능한 식으로 인식하여 결과값을 돌려준다.

F

false /bin/false

항상 실패

주로 쉘스크립트에서 사용하며 아무것도 하지 않고 항상 《실패》 또는 《거짓》을 반환한다.

형식 : false,false{--help, --version}

-help : 사용법을 출력한다. -version : 판본정보를 출력한다.

상세한 사용법은 쉘스크립트에서 설명한다.

참 고

xhos를 활용하기

2대이상의 콤퓨터를 사용하는 경우에 하나의 호스트에서 다른 하나의 호스트의 X 프로그람들을 실행시켜 사용할수 있다. 매 콤퓨터의 IP를 192.168.0.2와 192.168.0.3이라고 가정하자.

먼저 192.168.0.3의 호스트에서 192.168.0.2의 호스트에로 원격접속한다.

```
# ssh root@192.168.0.1
```

접속후 DISPLAY환경변수를 192.168.0.3:0.0으로 설정하여 이후에 실행되는 프로그람들을 192.168.0.3의 호스트에서 보여지도록 한다.

```
# export DISPLAY=192.168.0.3:0.0
```

시험적으로 kcctongbo창을 실행시켜보자.

```
# kcctongbo &
```

192.168.0.2호스트의 kcctongbo창을 192.168.0.3의 호스트에서 사용할수 있다.

fdformat

/usr/bin/fdformat

플로피디스크의 초기화

Linux에서 플로피디스크를 초기화하는 명령문이다.

형식 : fdformat[-n]장치이름

-n : 초기화를 진행한 다음 디스크표면검사를 하지 않는다.

Linux에서 플로피디스크를 가리키는 장치명은 /dev/fd0, /dev/fd1이다.

초기화를 하려면 용량을 지정해야 하는데 만일 1.44MB의 플로피디스크인 경우에는 /dev/fdoH1440으로 장치명을 초기화해야 한다. 아래의것은 장치이름을 보여준것이다. 표준 Linux에서 플로피디스크를 가리키는 장치명은 /dev/fd0, /dev/fd1이다. 현재 체계에 인식되여있는 장치명을 확인하려고 할 때에 제일 쉽게는 dmesg명령으로 확인해볼수 있다.

```
kcc@linux:~
# dmesg | grep floppy
Floppy drive(s): fd0 is 1.44M
```

현재 체계에 인식되여있는 플로피디스크장치명은 fd0이므로 이것을 초기화해보자.

```
kcc@linux:~
# fdformat /dev/fd0
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 KB.
Formatting ...done
Verifying ... Problem reading cylinder O, expected 18432, read 8192
```

플로피디스크를 넣고 /dev/fd0을 초기화한다.

플로피디스크장치가 1.44MB용이므로 /dev/fd0H1440을 지정하여도 된다.

```
kcc@linux:~
# fdformat /dev/fd0H1440
Duuble-szied, 80 tracks, 18 sec/track. Total capacity 1440 kB.
Formatting...done
Varifying...done
```

#ls /dev/fd0* 명령으로 장치들을 확인해보면 용량에 따라 아래와 같은 장치이름들을 볼수 있다. 이것들은 모두 플로피디스크에 해당하며 대부분의 경우에 사용하는것은 /dev/fd0H1440이다.

```
kcc@linux:~
# ls /dev/fd0*
/dev/fd0          /dev/fd0h1200  /dev/fdoh720    /dev/fd0u1743
/dev/fd0u3840
/dev/fd0CompaQ   /dev/fd0h1440  /dev/fd0h880    /dev/fd0u1760
/dev/fd0u720
/dev/fd0D360      /dev/fd0h1476  /dev/fd0u1040  /dev/fd0u1840
/dev/fd0u800
/dev/fd0D720      /dev/fd0h1494  /dev/fd0u1120  /dev/fd0u1920
/dev/fd0u820
/dev/fd0H1440    /dev/fd0h1660  /dev/fd0u1440  /dev/fd0u2880
/dev/fd0u830
/dev/fd0H360  /dev/fdoh360   /dev/fd0u1660  /dev/fd0u3200
/dev/fd0H720  /dev/fd0h410   /dev/fd0u1680  /dev/fd0u3520
/dev/fd0d360   /dev/fd0h420  /dev/fd0u1722  /dev/fd0u360
```

fdisk

/sbin/fdisk

구획설정

구획(Partition)을 설정하는 명령문이다.

형식 : fdisk[설정값][장치이름]

-v : 판본정보를 보여준다.

-l : 현재 체계의 구획표(Partition Table)를 보여준다.

-s 구획 : DOS구획이 아닌 한 구획의 크기를 보여준다.

Linux를 설치할 때에 구획설정은 보통 Disk Druid와 fdisk, cfdisk로 하게 된다.

Disk Druid를 리용하면 대부분 간단히 설정할수 있으나 구획에 관한 보다 자세한 설정을 할 때에는 fdisk를 리용해서 구획설정을 해야 한다.

먼저 설치되여있는 Linux에서 /dev/hda 장치를 fdisk명령으로 보도록 하자.

fdisk -l명령을 리용하면 지정한 구획의 정보를 볼수 있다.

```
kcc@linux:~
# fdisk -l /dev/hda
Disk /dev/hda: 255 heads, 63 sectors, 1826 cylinders
Units=cylinders of 16065 * 512 bytes
DeviceBoot  Boot  Star    End        Blocks   Id
System
/dev/hda1   *1    7       56196      83       Linux
/dev/hda2   8     20      104422+    83       Linux
/dev/hda3   21    339     2562367+   83       Linux
/dev/hda4   340   1826    11944327+  5        Extended
/dev/hda5   340   372     265041     82       Linux swap
/dev/hda6   373   405     265041     82       Linux swap
/dev/hda7   406   788     3076416    83       Linux
/dev/hda8   789   1826    8337703+   83       Linux
```

-s 설정값을 리용하면 지정한 구획의 크기를 볼수 있다.

```
kcc@linux:~

# fdisk -s /dev/hda
14668416
# fdisk -s /dev/hda1
56196
# fdisk -s /dev/hda2
104422
```

그러면 직접 fdisk를 통해 /dev/had구획을 설정하는 사용법을 보기로 하자.

```
kcc@linux:~

# fdisk /dev/hda
The number of cylinders for this disk is set to 1826.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time(e.g.,old versions of LILO
2) booting and partitioning software from other OSs
   (e.g.,DOS FDISK,OS/2 FDISK)

Command (m for help):
```

m을 입력하면 사용법을 볼수 있다.

```
kcc@linux:~

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partiton
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
```

```
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition s system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality(experts only)
Command (m for help):
```

p설정값을 리용하면 현재 구획표를 볼수 있다.

```
kcc@linux:~

Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 1826 cylinders
Units=cylinders of 16065 * 512 bytes
Device      Boot   Start   End         Blocks   Id  System
/dev/hda1   *1     7       56196       83           Linux
/dev/hda2    8     20      104422+     83           Linux
/dev/hda3    21    339     2562367+    83           Linux
/dev/hda4    340   1826    11944327+   5            Extended
/dev/hda5    340   372     265041      82           Linux swap
/dev/hda6    373   405     265041      82           Linux swap
/dev/hda7    406   788     3076416     83           Linux
/dev/hda8    789   1826    8337703+    83           Linux
```

command(m for help):에 입력할수 있는 명령문

- a: 기동가능한 flag로 변경
- b: bsd디스크표를 편집
- c: DOS호환의 flag로 변경
- d: 구획삭제
- I: 알려진 구획형태의 목록
- m: 이 안내문을 출력
- n: 새로운 구획을 추가

- o: 새로운 DOS구획표를 생성
- p: 구획표를 출력
- q: 변경을 보관하지 않고 완료
- s: 새로운 Sun디스크표를 생성
- t: 구획의 체계ID를 변경
- u: 표시/entry단위를 변경
- v: 구획표를 검사
- w: 디스크에 표를 기록하고 탈퇴함
- x: 다른 기능(숙련자만이 리용할것)

구획형식의 코드 및 종류

0 비여있음	1b숨겨진 Win95 FAT	64 Novell Netware	bb숨겨진 기동위자드
1 FAT12	1c숨겨진 Win95 FAT	65 Novell Netware	C1 DRDOS/Sec
2 XENIX root	1e숨겨진 Win95 FAT	70 DiskSecure Mult	C4 DRDOS/see
3 XENIX usr	24 NEC DOS	75 PC/IX	C4 DRDOS/sec
4 AT16 <32M	39 Plan 9	80 구형 Minix	c7 Syrinx
5 확장	3c PartionMagic	81 Minix/구형Linux	Da Non-FS Data
6 FAT16	40 Venix 80286	82 Linux swap	db CP/M/CTOS
7 HPFS/NTFS	41 PPC PreP Boot	83 Linux	de Dell 편의프로그람
8 AIX	42 SFS	84 OS/2숨겨진 C:	df Bootit
9 AIX boot가능	4d QNX4x	85 Linux extended	e1 DOS access
a OS/2 boot 관리자	4e QNX4.x 2번째	86 NTFS volum set	e3 DOS R/O
b Win95FAT32	4f QNX4.x 3번째	87 NTFS volum set	e4 SpeedStor
c Win95FAT32	50 QnTrack DM	8e Linux LVM	eb BeOS 파일체계
e Win95FAT16	51 QnTrack DM6 Aux	93 Amoeba	ee EFI GPT
f Win95Ext'd	52 CP/M	94 Amoeba BBT	ef EFI (FAT-12/6/ 32)
10 OPUS	53 OnTrack DM6 Aux	9f BSD/OS	f1 SpeedStor
11 숨겨진 FAT12	54 OnTrack DM6 a0	IBM Thinkpad hi	f4 SpeedStor
12 Compaq diagnost	55 EZ-Drive	a5 BSD/386	f2 DOS Secondary

14 숨겨진 FAT16	56 Gdden Bow	a6 OpemBSD	fd Linux raid auto
16 숨겨진 FAT16	5c Priam Edisk	a7 NexTSTEP	fe LANstep
17 숨겨진 HPFS/NT	61 SpeedStor	b7 BSDI 파일체계	ff BBT
18 AST Windows	63 GNU HURD	b8 BSDI swap	

그러면 자세한 fdisk사용법을 실지 례를 들어 보기로 하자. 20GB하드디스크를 실례로 한 경우, 크게 여러가지 구획설정형태가 있지만 아래와 같은 2가지의 경우로 보기로 하자.

1. Windows 6GB, Linux 6GB, 나머지 령역은 Windows예비보관령역으로 설정하는 경우

① Windows상에서 fdisk를 입력하여 약 6GB정도 설정후에 설치한다.
(/dev/hcd1에 6GB용량의 구획을 설정)

② Linux설치과정중에 구획설정에서 fdisk로 들어간다.

③ 《p》를 입력하여 구획정보를 본다.

④ hda1을 제외한 모든 령역을 삭제한다.

⑤ 새로운 구획생성후, 구획형식을 지정한다.

n → p → 2→ →+6000MB

⑥ /dev/hda3령역에 128MB의 swap를 생성한다.

n → p → 3 → →+128MB → t → 82

⑦ /dev/hda4령역에 나머지의 남은 디스크용량을 모두 예비보관령역으로 만든다.

n → p → 4 → → → t → c

⑧ 구획설정보관(w)후 Linux를 설치한다.

2. 20GB하드디스크에 모두 Linux를 설치하는 경우

① Linux설치과정중에 구획설정에서 fdisk로 들어간다.

② 《P》를 입력하여 구획정보를 본후 모두 삭제한다.

③ /dev/hda1령역에 128MB의 swap를 생성한다.

n → p → 1 → → +128M → t → 1 → 82

④ /dev/hda2 령역에 모든 나머지용량을 Linux native령역으로 생성한다.

n → p → 2 → →

⑤ 구획설정보관(w)후 Linux를 설치한다.

【관련명령문】

cfdisk : 유표안내방식의 구획설정명령문

fetchmail /usr/bin/fetchmail

우편을 모아오기

봉사기에서 우편을 검색하여 가져오는 편의프로그람이다.

형식 : fetchmail[설정값][우편봉사기]

-V -version : 판본정보를 보여준다.

-c -check : 실지로 우편을 가져오거나 삭제하지 않고 대기하는 상태

-s -sillent : 통보문을 보여주지 않는다.

-v -verbose : 진행상황을 자세히 보여준다.

-u -usermame : 우편봉사기에 접속할 사용자식별이름을 지정한다.

-p -protocol : 통신규약을 지정한다. 아래의 통신규약들이 지정가능하다. AUTO, POP2, POP3, APOP, KPOP, SDPS, IMAP, ETRN, ODMR

-k -nokeep : 우편을 검색한 후에 새로운 통보문을 봉사기상에서 삭제한다.

-k -keep : 우편을 검색한 후에 새로운 통보문을 봉사기상에서 보관한다.

fetchmail은 POP2, POP3, IMAP2bis, IMAP4, IMAPrev1 등의 통신규약을 통해 봉사기에서 전자우편을 가져오며 통보문은 국부령역(Local)에 있는 sendmail을 통해 25번 포구(Port 25)로 전달하게 된다.

먼저 시험을 위해 우편을 mail명령을 리용하여 통보문을 보낸다.

```
kcc@linux:~
# mail admin@linuxroot.co.kp
Subject: fetchmail testing
test
.
Cc: #fetchmail -p pop3 -u admin -k mail.linuxroot.co.kp
Enter password for admirn@mail.linuxroot.co.kp:
fetchmail: No mail for admin at mail.linuxroot.co.kp
You have new mail in /var/spool/mail/root
```

통신규약을 POP3으로 지정하며 검색된 우편은 POP3봉사기에 보관하여둔다. 확인결과 새로운 통보문이 /var/spool/mail/root파일로 보관된것을 볼 수 있다.

참 고

우편봉사기에 있는 우편을 다른 식별이름으로 옮기기.

Linux봉사기에 도착해있는 우편을 다른 우편식별이름(Mail Account)으로 옮기려고 할 때에 우편을 손쉽게 옮겨주는 방법이다. 만일 linuxroot.co.kp봉사기의 admin에 온 우편을 jkwoo식별이름으로 옮기려고 하는 경우 다음과 같이 명령을 실행한다.

우편봉사기에 도착하는 우편은 /var/spool/mail 등록부의 아래준위에 있는 매 식별이름등록부에 축적되게 된다.

```
# cat/var/spool/mail/admin | formail -s/usr/sbin/sendmail jkwoo@linuxroot.co.kp
```

이와 같은 작업을 자동적으로 하기 위해서는 다음과 같이 .forward 라는 숨김파일을 만들어 둔다. 자신에게 오는 모든 우편을 jkwoo@kccredstar.co.kp라는 식별이름으로 보낼 때에는 .forward라는 이름의 파일을 만들고 이 우편주소만 적어놓으면 된다. 동시에 여러 식별이름으로 보낼 때에는 다음과 같이 입력하고 보관한다.

```
jkwoo@kccredstar.co.kp
happy@happylinuxing.com
rurulala@ kccbook.co.kp
```

file

/usr/bin/file

파일류형확인

파일의 류형을 파일정보자료파일에서 검색하여 반환한다.

형식 : file[설정값][파일]

-b : 지정한 파일명은 보여주지 않고 파일의 류형만을 보여준다.

-f 파일목록 : 파일목록에서 지정한 파일들에 대해 지령을 실행한다.

-i : 보통의 보기 쉬운것보다 MIME형식을 보여준다(례를 들면 ASCII text를 text/plain; chatset-us-ascii형태로 보여준다).

-m magic file : 기정으로 지정된 /etc/magic을 사용하지 않고 magic file을 지정한다.

-L : 기호련결된 파일을 추적하여 시도한다.

-v : 판본정보를 보여준다.

file명령문은 파일류형을 구분하는 편의프로그람으로서 file명령뒤에 파일이름을 주면 ascii, text, date와 같이 파일이름의 류형을 알려준다.

/usr/share/magic 혹은 /usr/lib/magic이라는 파일에 여러 파일에 대한 정보를 가지고있어 파일의 류형을 알수 있다.

아래의 실례는 test파일에 대한 파일류형을 알려준다.

```
kcc@linux:~

# file test
test1: ASCII text
```

-b 설정값을 사용하면 보관한 파일이름은 제외하고 류형만을 볼수 있다.

```
kcc@linux:~

# file -b test
ASCII text
```

-i설정값을 사용하면 MIME형식을 볼수 있다.

```
kcc@linux:~

# file -i test1
file: Using regular magic file /usr/share/magic.mime
test1: text/plain; charset=us-ascii
```

참 고

MIME

ASCII자료만을 처리할수 있는 원래의 인터네트 전자우편통신규약인 SMTP를 확장하여 음향, 정지화상, 동화상, 응용프로그람 등 여러가지 종류의 자료파일들을 주고 받을수 있도록 기능을 확장한 통신규약이다.

find

/usr/bin/find

파일찾기

주어진 조건에 따라 등록부의 가지를 검색하여 해당한 파일을 찾는 명령문이다.

형식 : find[경로][설정값][작업]

-name name : 지정된 이름의 파일을 찾는다.
-user name : user소유의 파일을 찾는다.
-type [bcdfls] : 지정된 형식의 파일을 찾는다.
  b : 블로크파일
  c : 문자
  d : 등록부
  f : 파일
  l : 련결파일
  s : 소케트
-size[+/-]n[bokw] : 지정된 크기의 파일을 검색한다.
  ＋n : n보다 크다.
  －n : n보다 작다.
  n : n이다
  b : 512-byte
  c : byte
  k : kilobytes
  w : 2-byte
-inum number : 지정한 inode번호와 파일을 검색한다.
-print : 표준출력으로 검색된 파일명을 출력한다.
-exec command {}\; : 찾은 각 파일에 대해 지정된 지령을 실행한다.
-ok command {} \; : 실행여부를 사용자에게 확인한 후 지령을 실행한다.

find명령문은 복잡한 파일체계구조를 가지는 Linux에서 파일을 찾아내는 명령문이다. 실지로 find는 파일검색보다 선택조건에 일치하는 파일을 찾아서 명령을 실행하는 파일체계로 사용된다.

이름으로 찾는 find사용법과 명령행을 리용한 find사용법을 보기로 하자.

find는 검색할 경로를 미리 정의해준후에 설정값과 사용법이 첨부되게 된다.

뿌리등록부의 word.awk를 검색하여보자. 이름을 기준으로 하여 검색하는 경우에 -name확장자를 사용해야 한다.

```
kcc@linux:~
# find /root -name word.awk
/root/word.awk
```

exec command \;는 find가 찾아낸 매 파일에 대해 command를 실행한다. command는 \;로 끝나야 한다.

ㄱ) 현재 등록부에 있는 모든 파일이름을 라렬하여 보여주기

```
# find . -print
```

ㄴ) 현재 등록부에 있는 등록부들을 라렬하여 보여주기

```
# find . -typed -print
```

ㄷ) 현재 등록부에서 파일이름에 공백문자가 들어간 모든 파일을 삭제하기

```
# find . -name "* *" -exec rm -f {} \;
```

[주의]

+ { ; " \ = ? ~ ( ) < > & * | $ 등의 특수기호를 포함한 파일을 삭제한다. 이 과정에서 2>/dev/null 오유출력은 무시한다.

ㄹ) 현재 등록부파일들중 파일이름에 비정상의 문자를 포함한 파일을 지우기

```
kcc@linux:~
# for filename in *
>do
>badname=echo "$filename" | sed -n >/[\+\{\;\" \\\=\?~\(\)\<\>\&\*\|$}/p
>rm $badname 2>/dev/null
>done
```

ㅁ) 원하는 파일체계를 모두 복사하기

```
# mkdir /hdc1
# mount /dev/hdc1 /hdc1
# cd/home
# find . -depth -print | cpio   pmdv /hdc1
```

ㅂ. 지난 24시간동안에 변경된 모든 파일목록을 changelist파일에 만들기

```
# find / -mtime -1\! -type d -print > changelist
```

finger

/usr/bin/finger

사용자정보보기

사용자정보를 찾아보는 명령문이다.

형식 : Finger[-lmsp][user…][user@host…]

-l : 다중행형식으로 사용자의 home등록부, 집전화번호, 체계접속쉘, 우편상태, home등록부파일 등과 함께 s설정값입력에 의한 정보를 보여준다.

-s : 사용자의 체계접속(login)이름, 실지이름, 말단이름, 상태, 유휴(idle)시간, 체계접속시간, 사무실위치, 사무실전화 등을 보여준다.

finger는 특정사용자정보를 알려주는 기능을 한다.

이것은 편리한 기능을 제공하지만 악의를 가진 침입자에게는 유용한 정보로 될수 있으므로 될수록이면 finger설정을 하지 않는것이 좋다. RedHat 7.2의 경우, finger설정은 /etc/xinetd.d/finger파일에서 한다.

```
kcc@linux:~
# cat /etc/xinetd.d/finger
 service finger
 {
         disable=no
         socket_type     =stream
         wait            =no
         user            =nobody
         server          =/usr/sbin/in.fingerd
 }
```

disable=no로 설정한 다음 xinetd데몬을 다시 실행한다.

/etc/xinetd.d/finger파일을 수정하였다면 다음과 같이 xinetd데몬을 다시 시작한다.

```
kcc@linux:~

# /etc/rc.d/init.d/xinetd restart
Stopping xinetd:                                      [ OK ]
Starting xinetd:                                      [ OK ]

# finger
Login   Name   Tty    Idle   Login Time    Office   Office Phone
root    root   *:0           May 10 17:24
root    root   pts/1  8:51   May 13 10:45
root    root   pts/3         May 13 20:51(211.49.155.249)

# finger -l
Login: root                          Name: root
Directory: /root                     Shell:/bin/bash
On since Fri May 10 17:24(KST) on :0 (messages off)
On since Mon May 13 10:45 (KST) on pts/1  9 hours 3 minutes idle
On since Mon May 13 23:53 (KST) on pts/3 from 211.49.155.249
New mail received Mon May 13 04:08 2004(KST)
    Unread since Tue May 7 13:29 2004(KST)
No Plan.

#finger -m
Login   Name   Tty    idle   Login Time    Office   Office Phone
root    root   *:0           May 10 17:24
root    root   pts/1  9:04   May 13 10:45
root    root   pts/3         May 13 23:53(211.49.155.249)

# finger admin
Login: admin                Name: admin
Directory: /home/admin      Shell:/bin/bash
Last login Fri Dec 14 13:01 (KST) on :0
No mail.
No Plan.
```

free

/usr/bin/free

체계주기억의 보기

체계주기억의 빈 령역과 사용한 주기억공간을 출력하는 명령문이다.

형식 : free[-b|-k|-m][-o][-s delay][-t][-v]

-b : 주기억공간을 byte단위로 표시한다.

-k : Kbyte단위로 표시한다.

-m : Mbyte단위로 표시한다.

-t : 총합계가 포함된 행을 출력한다.

-o : 완충기(buffer)에 조정된 행출력을 비할성화한다.

-s : delay초마다 계속해서 출력하도록 한다. delay에 임의의 류동소수점수도 사용할수 있다.

free명령문으로는 현재 주기억기(Memory)의 사용용량을 확인해볼수 있게 한다. 비여있거나 사용된 물리적인 주기억용량, swap주기억의 전체량 등을 보여준다. 또한 핵심부에 의해 사용되는 공유기억기와 완충령역도 보여준다.

```
kcc@linux:~

# free
          total    used     free    shared   buffers    cached
Mem:      126040   118580   7460    296      18424      29232
-/+ buffers/cache:          70924   55116
Swap:              530064   54792   475272
```

《free -s초》설정값을 주면 초변화로 일어나는 주기억사용상태를 알아볼 수 있다.

```
kcc@linux:~

# free -s 1
        total    used     free     shared buffers  cached
Mem:             126040   118532   7508   296      18424   29228
```

```
-/+buffers/cache: 70880    55160
Swap:             530064  54792   475272

             total     used      free      shared  buffers   cached
Mem:              126040  118532  7508     296      18424    29228
-/+buffers/cache: 70880    55160
Swap:             530064  54792   475272

             total     used      free      shared  buffers   cached
Mem:              126040  118532  7508     296      18424    29228
-/+buffers/cache: 70880    55160
Swap:             530064  54792   475272
```

참 고

합리적인 방법으로 swap구획을 확보하는 방법

swap구획(swap partition)을 설정할 때에 주기억기의 2~3배크기로 정하는것이 일반적이다. 물론 이와 같은 경험적인 방법도 리용할수 있지만 합리적이고 정확하게 swap를 할당하는 방법을 취하는것이 좋다. 왜냐하면 swap는 모자라서는 안되지만 너무 크게 정하면 랑비로 되기때문이다.

먼저 필요한 최대주기억을 확인한다. 최대주기억이란 한번에 구동하려고 하는 모든 프로그람이 요구하는 주기억기의 총용량이다. 례를 들어 X Window환경에서는 대략 8MB이상의 주기억이 필요하며 KDE환경에서는 더 많이 요구되므로 넉넉히 확보해주어야 한다.

gcc로 프로그람을 번역할 때에는 보통 4MB정도가 필요하지만 프로그람의 크기에 따라 수십MB이상의 주기억이 필요한 경우도 있다. 핵심부(kernel)는 그 자체가 1MB 정도를 차지하며 일반적인 쉘과 작은 편의프로그람 역시 모두 합쳐 1~2MB정도 차지한다. 또한 같은 프로그람을 동시에 여러개 실행하거나 혹은 서로 다른 사용자가 제각기 사용하고있는 경우에도 실행코드와 공유서고의 용량을 제외한 나머지 부분은 각각 계산하여주어야 한다.

이와 같이 산출된 용량에 여유분을 어느정도 더한다. 대체로 계산된 크기의 10~20% 정도를 여유분으로 정한다. 여기에서 실제 주기억의 용량을 빼면 남는 필요량이 swap의 합리적인 크기로 된다. 이러한 계산으로 얻어진 swap공간이 실제 주기억의 2~3배를 넘는 경우에는 주기억공간을 더 확장해주어야 한다. 그렇지 않으면 체계의 실행속도저하를 가져오게 된다.

fsck

/sbin/fsck

파일체계검사

파일체계를 검사하고 문제성있는 부분은 보수하는 명령문으로서 기동할 때에 자동적으로 파일체계를 검사한다.

형식 : fsck[-AVRTNP][-s][-t 파일체계형식] [파일체계설정값]파일체계[...]

-f : fsck는 파일체계에 이상이 없다고 판단하면 검사하지 않는데 이러한 때에 무조건 검사를 하도록 한다.

-A : /etc/fstab에 등록된 내용을 참고하여 체계에서 사용하는 모든 파일체계에 대해 검사를 진행한다.

-a : fsck가 검사를 진행하는중에 오유가 발생했을 때 그것을 자동적으로 처리하도록 한다. 검사진행중의 질문들에 일일이 대답하는것이 시끄럽거나 모두 y로 처리하여도 무방한 경우에 사용한다.

-b : super block가 파괴된 경우에 사용하는것이다. 이것은 블로크의 첫번째에 위치하는데 만일 super block가 파괴된 경우에는 매 8192블로크마다 예비보관된 super block의 복사본을 가지고 super block를 복구해야 한다.

-R : -A plug와 같이 사용될 때에 root파일체계는 제외하도록 한다(이미 읽기 및 쓰기가능으로 설치(mount)되여있는 경우).

-T : 시작할 때에 제목을 보여주지 않는다.

-N : 실행하지는 않고 어떤 작업을 할것인가만을 보여준다.

-P : A plug가 설정되여있을 때에 다른 파일체계와 root파일체계를 병렬처리하여 검사한다. root파일체계에 오유가 있는 경우, e2fsck 2진값이 파괴되여있을수 있으므로 안전한 방법은 못된다. 체계관리자들을 위해서만 제공되는 기능이다.

-s : fsck동작을 순차화한다. 대화형방식에서 여러 파일체계를 검사할 때에 유용하다. (주의: e2fsck은 주로 대화형으로 실행한다. e2fsck를 비대화형방식으로 실행하려면 -p 또는 -a설정값을 주어 오유를 자동퇴치하거나 -n을 주어 고치지 않고 그대로 두어야 한다.)

-V : 실행되는 각 파일체계용의 지령을 포함하여 자세한 출력을 진행한다.

-t 파일체계류형 : 검사할 파일체계류형을 지정한다. A flag가 설정되여있는 경우, 파일체계류형에 맞는것만 검사한다. 파일체계류형앞에 no를 붙이면 파일체계류형이 아닌것만 검사한다. 보통 /etc/fstab파일에 들어있는 파일체계류형을 따른다. 만약 해당한 류형을 알아낼수 없는 경우에 fsck는 -t설정값에 주어진 값을 사용한다. 주어진 값이 유효하지 않은 경우에 현재의 기본값인 ext2가 사용된다.

거의 모든 경우에 Linux체계는 기동시에 자동적으로 파일체계를 검사하게 되는데 그것은 /etc/rcd/rcsysinit파일에서 fsck설정을 해놓았기때문이다. 그러므로 수동으로 fsck로 검사하는 경우는 많지 않다. 체계를 기동할 때마다 파일체계를 검사하고 자동복구하게 된다. 만일 다른 등록부가 참고하지 않는 활성 inode가 발견되면 /lost+found 등록부를 사용하며 파괴된 파일은 /lost+found 등록부에 존재하게 된다.

/dev/hdc1 파일체계를 강제로 수행하고 매 파일체계용의 명령을 포함하여 자세한 출력을 나타내도록 해보자.

```
kcc@linux:~

# fsck -f -v/dev/hdc1
Parallelizing fsck version 1.23(15-Aug-200l)
e2fsck 1.23, 15-Aug-2001 for EXT2 FS 0.5b, 95/08/09
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
    5766 inodes used (0%)
    107 non-contiguous inodes (1.9%)
    # of inodes with ihd/dind/tind blocks: 1696/155/0
1293868 blocks used (81%)
        0 bad blocks
        0 large files

4917 regular files
```

```
      828 directories
        0 character device files
        0 block device files
        0 filos
        0 links
       12  symbolic links (12 fast symbolic links)
        0 sockets
--------
  5757 files
```

【관련명령문】

ext2 : 파일체계검사. tsct -t ext 명령과 같다.

참 고

새로운 하드디스크의 설치

현재 사용하고있는 하드디스크는 Primary Master(/dev/hda)에 설치되여있으며 추가할 하드디스크는 Primary Slave(/dev/hdb)에 설치하였다고 가정한다.

fdisk실행

새 하드디스크에 Linux구획을 설정한다.

```
# fdisk /dev/hdb
```

구획정보확인

구획의 정보를 확인한다. 새로운 장치의 설치때에 아래와 같이 구획정보가 나타나지 않는다.

```
Command (m for help): p
Disk /dev/had: 255 heads, 63 sectors, 788 cylinders
Units=cylinders of 16065 * 512 bytes
Units=cylinders of 16065 * 512 bytes
Device Boot Start End Blocks Id System
```

새로운 구획생성

새로운 구획을 생성한다. 아래의 내용은 구획을 하나로 설정하는 경우이다.

```
Command (m for help): n
Command action
e extended p primary partition (1-4)
p
(구획을 Primary로 설정한다.)
Partition number (1-4): 1
(사용할 구획수자를 입력한다)
First cylinder (1-788, default 1):
Using default value 1
(초기값을 입력한다.)
Last cylinder or +size or +sizeM or +sizeK(1-788, default
788):
Using default value 788
(마지막 실린더값을 입력한다.)
```

새로운 구획정보확인

```
Command (m for help): p
Disk /dev/hda: 255 heads, 63 sectors, 788 cylinders
Units=cylinders of 16065 * 512 bytes
Device Boot Start End Blocks Id System
/dev/hdb1 1 788 6329578+83 Linux
```

보관후 완료

구획을 보관한 후에 완료한다.

```
Command (m for help): w
```

파일체계생성

재기동후에 분할한 구획을 ext2파일체계로 생성한다.

```
# mke2fs /dev/hdb1
```

또는 다음과 같은 방법을 사용할수도 있다.

```
# mkfs -t ext2 /dev/hdb1
```

mount

파일체계생성후 /hdb1에 mount한다.

```
# mkdir /hdb1
# mount /dev/hdb1 /hdb1
```

ftp

/usr/bin/ftp

ftp봉사기접속의뢰기프로그람

ftp봉사기에 접속하기 위한 ftp의뢰기이다.

**형식 : ftp[-v][-d][-i][-n][-g][-k realm][-f]
[-x][-u][-t][접속호스트명]**

-v : 자료전송통계 및 봉사기의 반응을 모두 보여준다.

제일 초기에 나온 ftp의뢰기프로그람으로서 대부분의 Linux배포판들에 들어 있다.

접속후 사용할수 있는 명령문들을 아래에 주었다.

- ftp>
- ? : 사용법에 관한 정보를 보여준다.
- acsii mode: ASCII와 같다.
- binary mode: binary와 같다.
- conect: 호스트[포구]지정한 호스트와 포구로 접속시도한다.
- get**파일이름**: 원격접속호스트의 파일을 국부체계에 파일내리적재를 한다.
- get **파일이름 국부파일이름**: 원격접속호스트의 파일을 지정한 국부파일이름으로 내리적재한다.
- get **파일이름1 파일이름2 파일이름3…파일이름N**: 지정한 파일들을 내리적재한다.
- mode **전송방식**: 전송방식(ASCII or binary)으로 방식설정한다.
- put **파일이름**: 국부파일을 원격접속호스트에 올리적재한다.
- put **국부파일이름 원격파일이름**: 국부파일이름을 원격접속호스트에 지정한 원격파일이름으로 올리적재한다.
- put **파일이름1 파일이름2 파일이름3… 파일이름N**: 원격등록부의 지정한 파일들을 지정한 원격등록부에 올리적재한다.
- quit: ftp접속을 끝낸다.

ftp봉사기접속방법

례를 들어 kccredstar.co.kp라는 호스트에 접근하는 방법에는 ftp와 호스트이름을 련속하여 입력하는 방법, ftp명령으로 해서 open명령으로 호스트에 접속하는 방법 등이 있다.

```
# ftp kccredstar.co.kp
```

또는 아래와 같이 사용할수도 있다.

```
# ftp
ft > open linuxroot.co.kp
```

ftp봉사기접속

접속시도를 하면 ftp봉사기프로그람을 알수 있고 접속하려고 하는 사용자식별이름정보를 입력하는 입력독촉상태(prompt)가 화면상에 나타난다.
만일 식별이름을 입력하지 않으면 anonymous식별이름으로 접속하게 된다.

```
kcc@linux:~
# ftp kccredstar.co.kp
Connected to kccredstar.co.kp.
220 ProFTPD 1.2.5rc1 Server ready.
500 AUTH not understood.
500 AUTH not understood.
KERBEROS_V4 rejected as an authentication type
Name(kccredstar.co.kp:root):
```

admin식별이름정보를 입력하고 암호를 입력하여 접속을 한다.

```
kcc@linux:~
# ftp kccredstar.co.kp
Connected to kccredstar.co.kp.
220 ProFTPD 1.2.5rc1 Server ready.
500 AUTH not understood.
500 AUTH not understood.
KERBEROS_V4 rejected as an authentication type
```

```
Name (kccredstar.co.kp:root): admin
331 Password required for admin.
Password:
230 User admin logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

파일전송형식

ftp파일전송형식의 기본방식은 ascii로서 ascii파일전송에 사용된다. 2진방식으로 전환하려면 bin명령을 입력한다.

```
kcc@linux:~
ftp > bin
200 Type set to I.
```

ftp봉사기목록보기

봉사기의 목록을 보기 위해서는 ls명령을 사용한다.

```
kcc@linux:~
ftp> ls
227 Entering Passive Mode(211,255,298,59,7,240).
150 Opening ASCII mode data connection for file list
drwxr-xr-x 3  admin   admin   4096  Dec  14  13:32   Desktop
drwxr-xr-x 2  admin   admin   4096  Dec  14  13:01   Desktop.back
-rw-r—r—  1  admin   admin   0          Feb  2 10:16  Recycled
-rw-r—r—  1  admin   admin   6688768       Jul   4  2001
SSHWinClient-3.0.0.exe
-rw-r—r—  1  admin   admin   0      Feb  2   10:16   _Restore
drwxr-xr-x 4  admin   admin   4096  Dec  14  13:01   kde. back
drwxrwxr-x  2  admin   admin   4096  Mar 26  11:33   movies
```

등록부이동

cd명령을 리용하면 등록부를 이동할수 있다.

《250》으로 시작하는 통보문은 봉사기가 보내는 정보통보문이다.

```
hanbit@linux:~
ftp> cd Desktop
250 CWD command successful.
```

파일받기와 보내기

받기는 《get 봉사기파일이름》을 사용하며 보내기는 《put 국부파일이름》을 사용한다. 만일 여러 파일을 받거나 보내기 위해서는 mget, mput명령을 사용한다.

```
kcc@linux:~
ftp > get test.tar.gz
local:tes.tar.gz remote: test.tar.gz
227 Entering Passive Mode (211,255,253,59,7,251).
150 Opening BINARY mode data connection for genie (285
bytes).
226 Transfer complete.
285 bytes received in 0.019 seconds (15 Kbytes/s)
```

현재의 국부 파일등록부를 변경하려면 !cd명령을 사용하도록 한다.

```
kcc@linux:~
ftp> !cd /home/file
Local directory now /home/file
```

【관련명령문】

ncftp : ftp봉사기접속의리기프로그람이다. 기본적으로는 전화면방식으로 동작하며 〈TAB〉건 완성기능을 사용할수 있다.

참 고

Hash mark

만일 ftp로 파일을 내리적재받는 동안 그의 진행을 보려고 하는 경우, hash명령문을 사용한다. ftp는 hash mark를 내리적재받는 자료의 1024byte마다 보여준다.

```
kcc@linux:~
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
```

ftpcount

usr/bin/ftpcount

ftp사용자확인

매 class의 현재 사용자수를 보여준다.

형식 : ftpcount[-V]

-V : 저작권과 판본정보를 출력한다.

ftpcount명령은 ftpaccess file에서 정의된 매 class에 대한 사용자의 수 또는 제한된 사용자의 현재 접속수와 ftp봉사기의 PID를 보여준다.

```
kcc@linux:~
# ftpcount
Master proftpd process 2594:
Service    class    -    1 user
```

참 고

대기렬(queue)

자료가 들어간 순서대로 제거되는 자료구조를 말한다. 이러한 순서를 가리켜 흔히 선입선출(FIFO: first in, first out)이라고 한다. 이와 반대로 탄창(Stack)이라는 자료구조는 제일 먼저 들어간 자료가 가장 나중에 제거되는 형식인데 이것을 후입선출(LIFO: last in, first out)이라고 한다.

ftpwho

/usr/bin/ftpwho

ftp접속자보기

ftp에 접속해 있는 사람과 접속주소를 확인한다.

형식 : ftpwho [-V]

-V : 저작권과 판본정보를 출력한다.

ftp봉사기를 운영하거나 개인적으로 Linux상에서 ftp봉사기에 접근할 때에 현 시점에서 체계에 접근한 ftp사용자를 확인하려는 경우가 있다. 이러한 경우에 ftpwho명령을 사용하면 간단하게 확인할수 있다. 아래의것은 매 ftp user에 대한 현재의 처리정보를 보여준다. admin식별이름이 접속해 있고 127.0.0.1의 IP로부터 접속하였다는 정보를 알수 있다.

```
kcc@linux:~

# ftpwho
Master proftpd process 2594:
2610      5m10s    proftpd:    admin      -    127.0.0.1:    IDLE
Service   class                           -    1 user
```

fuser

/usr/bin/fuser

프로세스식별

보안을 위해 프로세스를 식별하고 통제한다.

형식 : fuser[설정값]

어떤 파일이나 파일집합, 등록부에 접근하고있는 프로세스를 PID로 식별해준다.

-k설정값을 리용하면 해당 프로세스를 제거할수 있다. 이 명령문은 체계보안의 측면에서 아주 흥미있는 기능인데 주로 스크립트에서 사용되여 체계봉사에 대해 허가받지 않은 사용자의 접근을 금지하는데 쓰인다.

아래의 실례를 보자. mount된 cdrom이 《Resource busy》같은 오유통보문으로 umount실패하는 경우가 있다.

이러한 때에 -k설정값은 /mnt/cdrom등록부아래에 머무르는 해당 프로세스를 강제로 제거시키는 기능을 수행한다.

```
kcc@linux:~

# umount  /mnt/cdrom
umount: /mnt/cdrom : device is busy

# fuser -km /mnt/cdrom
```

참 고

TX

Transmit Data의 략자로서 망을 통한 자료전송을 나타낸다.

RX

Receive Data의 략자로서 망을 통한 자료수신을 나타낸다.

G

gated /usr/bin/gated

게트웨이경로지정

게트웨이를 설정하고 운영한다.

형식 : gated [설정값]

-c : 문장에 오유가 있는가에 관해 설정파일을 분석하고 오유가 없으면 /usr/tmp/gated_dump 파일에 내용을 보관한다.

-C : 구문에 오유가 있는가에 관해 설정파일을 분석한다.

-f configfile : /etc/gated.conf파일을 대신하여 지정한 설정파일로 교체한다.

-n : 핵심부경로표를 변경하지 않고 검사만 진행한다.

-N : 데몬을 활성화하지 않는다.

-t traceoption : gated시작 때에 지정한 traceoption설정값을 사용한다. 만일 지정하지 않으면 general으로 시작한다. 아래의것이 지정가능한 traceoption들이다.

- adv : 방책(poicy)블로크관리
- all : normal, poloicy, route, state, task, timer를 포함한다.
- general : normal과 route를 포함한다.
- iflist : 핵심부대면부목록
- normal : 표준통신규약단계
- parse : 문맥분석 및 해석
- policy : 들어오거나 내보낸 경로가 적용되는 단계
- route : 경로표를 변경
- state : 체계변경보기
- task : 체계작업 및 대면부
- timer : 시계(timer)사용법

gated는 RIP, RIPng, EGP, BGP, BGP4+, HELLO, IS-IS, ICMP, ICMPv6 및 SNMP통신규약에 대한 게트웨이경로지정기능을 제공한다. gated설정파일은 /etc/gated.conf파일이다. 이 파일을 설정한 실례를 보기로 하자.

```
kcc@linux:~

#
# Generic configuration file for use with RIP
# rip yes {
        interface all noripin noripout;
        interface eth0 ripin ripout version 2 broadcast;
};

import Proto rip
        all ;
        default restrict ;
};

export proto rip {
        proto direct :
        proto static metric 1;
};
```

【관련명령문】

gdc : gated의 설정명령문

참 고

게트웨이(gateway)

게트웨이는 망으로 들어가는 입구역할을 한다. 경로지정(routing)의 관점에서 볼때에 인터네트는 많은 게트웨이와 호스트들로 구성된 망이라고 할수 있는데 망사용자의 콤퓨터와 웨브페지, 파일 등을 제공하는 봉사기는 호스트마디(host node)에 속하며 일반 기관, 기업소의 망에서 통신량(traffic)을 통제하는 콤퓨터나 인터네트봉사제공자콤퓨터는 게트웨이마디(gateway node)에 속한다. 게트웨이마디역할을 하는 콤퓨터가 대리봉사기(proxy server)나 방화벽봉사기(firewall server)의 역할을 함께 수행하는 경우도 있다.

gdc

/usr/bin/gdc

gated관리

gated관리를 위한 TCP/IP 명령문이다.

형식 : gdc[설정값]명령

-c coresize : 지정한 크기로 core dump의 최대크기를 설정한다.

-f filesize : 지정한 크기로 최대파일크기를 설정한다.

-m datasize : 지정한 크기로 최대자료토막크기를 설정한다.

-n : 핵심부회송표(kernel forwarding table)의 변경을 하지 않는다.

-q : 경고나 리력기록오유를 표준출력으로 보여주지 않고 syslogd로 리력기록된다.

-s stacksize : 지정한 크기로 최대의 stack크기를 설정한다.

-t seconds : 지정한 초로 끝내기 및 시작때에 완료되기를 기다리는 시간을 설정한다.

gdc는 gated에 설정관리가 가능하도록 한다. gdc에서 사용하는 명령문들을 아래에 주었다.

backout	이전 파일인 /etc/gated.conf: 파일에서 /etc/gated.conf파일로 보관하면서 만일 /etc/gated.conf: 파일이 없거나 빈 파일이면 실행을 하지 않는다.
BACKOUT	/etc/gated.conf:파일이 있든 없든 관계없이/etc/gated.conf파일로 보관한다.
checkconf	/etc/gated.conf: 설정파일의 문법을 검사한다.
checknew	새로운 파일인/etc/gated.conf+의 문법을 검사한다.
COREDUMP	gated의 dump core파일을 만든다.
createconf	새로운/etc/gated.conf+의 빈 파일을 생성한다. 이것은 소유자가 root, 파일집단이 gdmaint로 생성된다.
dump	gated에서 현재의 상태를 /user/tmp/gated_dump에 dump 한다.
interface	설정을 다시 읽어 들인다.
KILL	dated를 완료시킨다.
modeconf	모든 설정 파일을 mode 644의 소유자 root, 집단을 gdmaint로 설

	정한다.
newconf	/etc/gated.conf+파일을 /etc/gated.conf로 생성한다.
reconfig	설정파일을 다시 읽어들인다.
restart	gated를 중지후 다시 시작한다.
rmcore	core파일을 삭제한다.
rmdump	dump파일을 삭제한다.
rmparse	parse 오유파일을 삭제한다.
running	gated가 현재 실행중이면 상태 0으로 완료하고 그렇지 않으면 0이 아닌 상태로 완료한다.
start	gated를 시작한다.
stop	gated를 가능하면 순차적으로 정지시킨다.
term	순차적으로 완료시킨다.
toggletrace	gated추적을 실행 및 정지시킨다.
version	판본정보를 보여준다.

gdc명령은 gated봉사기 데몬설정을 위한 명령이다. 이것은 봉사기를 구동 및 정지시키며 구성설정파일의 유지보수 및 구문을 검사하고 수행상태를 검사할수 있다. gated 봉사기를 위한 설정파일인 /etc/gated.conf+를 생성하는 방법을 아래에 주었다.

```
kcc@linux:~
# gdc createconf
gdc: the "createconf" command requires group gdmaint to exist
Permission denied
```

새로운/etc/gated.conf+를 생성해보자. gdmaint집단이 없으므로 생성이 되지 않는다.

```
kcc@linux:~
# vi /etc/group
gdmaint:x:6000
"/etc/group" 55L, 760C written
```

/etc/group 파일을 열어 gdmaint집단을 추가하자.
다시 gdc creatconf 명령으로 생성된 파일을 볼수 있다.

```
kcc@linux:~
# gdc creatconf
# ls -al /etc/gated.conf+
-rw-rw-r--  1 root   gdmaint  0  7월   6 20:22  /etc/gated.conf+
```

【관련명령문】

gated : 게트웨이를 설정하고 운영하는 봉사기

getkeycodes

/usr/bin/getkeycodes

코드대응표보기

주사코드(Scan Code)와 건코드(Key Code)사이의 대응표를 보여주는 명령문이다.

주사코드와 건코드사이의 변환에는 setkeycodes편의프로그람을 사용한다. 이 기능을 사용하는 사람은 아주 적다. 건코드를 문자, 문자렬, 동작 등의 건대응으로 변환하는데는 loadkeys와 setmetamode편의프로그람을 사용하면 된다.

```
kcc@linux:~

# getkeycodes
Plain scancodes xx (hex) versus keycodes (dec)
0 is an error: for 1-88 (0x01-0x58) scancode equais keycode
0x58:   88   124   125   126   127     0     0     0
0x60:    0     0     0     0     0     0     0     0
0x68:    0     0     0     0     0   122     0   123
0x70:    0     0     0    89   120     0     0    90
0x78:   91    92    93    94    95   124   121     0
Escaped scancodes e0 xx (hex)
e0 00:    0     0     0     0     0     0     0     0
e0 08:    0     0     0     0     0     0     0     0
e0 10:    0     0     0     0     0     0     0     0
e0 18:    0     0     0     0    96    97     0     0
e0 20:    0     0     0     0     0     0     0     0
e0 28:    0     0     0     0     0     0     0     0
e0 30:    0     0     0     0     0    98     0    99
e0 38:  100     0     0     0     0   113   114   115
e0 40:  116   117     0     0     0     0   101   102
e0 48:  103   104     0   105   124   106   118   107
e0 50:  108   109   110   111     0     0     0     0
e0 58:    0     0     0   125   126   127     0     0
e0 60:    0     0     0     0     0     0     0     0
e0 68:    0     0     0     0     0     0     0   112
e0 70:    0     0     0     0     0     0     0     0
e0 78:    0     0     0     0     0     0     0     0
```

grep /bin/grep

문자렬검색

입력내용에서 특정문자렬을 검색하여 그 문자렬이 들어 있는 행을 화면에 출력하는 명령문이다.

형식 : grep[설정값] 패턴 [파일....]

-A num, -after-context=num : 지정한 num행만큼 일치하는 행뒤의 내용까지 보여준다.

-b, -byte-offset : 매 행의 결과를 보여주기전에 입력 파일의 byte -offset를 보여준다.

-B num, -before-context=num : 지정한 num행만큼 일치하는 행앞의 내용까지 보여준다.

-c, -count : 일치하는 행수를 보여준다.

-C[num], -before-context=num : 지정한 num행만큼 일치하는 행의 우와 아래의 내용을 보여준다. 기본은 두 행이다.

-d action, -directories=action : 등록부인 경우에 지정한 action을 실행한다. 기본 action은 read이다. 실행가능한 action은 다음과 같다.

- read : 등록부를 보통의 파일과 같이 읽는다.
- skip : 등록부를 건너뛴다.
- recurse : 등록부아래의 모든 파일들을 읽는다.

-e pattern, -regexp=pattern : 하나이상의 탐색패턴을 지정한다.
단순한 패턴과 같이 동작하지만 패턴이 -로 시작할 때에 유용하다.

-f file, -file=file : 패턴을 지정한 파일목록에서 가져온다.

-h, -no-filename : 패턴의 결과로 나오는 결과의 목록은 보여주지만 파일명은 보여주지 않는다.

-i, -ignore-case : 대소문자의 구별을 하지 않지 않는다.

-l, -files-with-matches : 일치하는 행의 파일명만 보여주고 행의 내용은 보여주지 않는다.

-n, —line-number : 일치하는 행의 결과와 그 파일에서와 결과 행이 몇번째 행인가를 보여준다.

-q, -quit, -silent : 결과를 보여주지 않는다.

-r, -recursive : 매 등록부의 하위파일들을 읽는다. -d recurse 설정값과 같다.

-s, -no-message : 존재하지 않거나 읽을수 없는 파일에 대한 결과의 오유를 보여주지 않는다.

-v, -revert-match : 지정한 패턴과 일치하지 않는것들을 보여준다.

-w, -word-regexp : 한 단어로 일치하는 결과만을 보여준다.

-x, -line-regexp : 패턴과 일치하는 전체 행수만 보여준다.

-y : -i 설정값과 동일하다.

grep는 본문파일내용을 검색하여 목적하는 문자렬패턴을 찾는 명령문이다. 자세한 사용법은 쉘스크립트에서 설명한다.

【관련명령문】

egrep=grep -E

fgrep=grep -F agrep grep가 류사정합(approximate matching)을 할수 있게 확장한다.

참　고

현재 등록부에 있는 등록부만 보기

이것을 alias로 만들어 편리하게 사용하도록 할수도 있다.

```
# is -la | grep "^d"  숨은 등록부까지 보기
# is -l | grep "^d"   보통 등록부만 보기
```

압축된 파일에서 검색

```
alias la='ls -la | grep "^d" '
```

zerep, zegrep, zigrep를 사용할수 있다. 이 명령문들은 압축되지 않은 파일에 대해서도 사용할수 있는데 grep, egrep, fgrep보다는 약간 실행속도가 느리다. bzip로 압축된 파일에서 검색하는 명령문으로서는 bzgrep가 있다.

groupadd

/usr/sbin/groupadd

집단추가

필요에 따라 새로운 집단을 만든다.

형식 : groupadd[설정값]집단

-g gid : 지정한 gid로 집단 ID를 생성한다.

-o : -g설정값과 함께 사용하며 지정한 gid가 이전에 있어도 중복을 허용한다.

-r : 체계관리령역인 499이하의 집단 ID에서 집단을 추가한다.

groupadd 명령은 새로운 집단을 추가하는 명령으로서 /etc/group과 관련이 있다.

```
kcc@linux:~
# cat/etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:
-----------중간 생략--------------
dba:x:5000:oracle:root
```

dba집단에는 oracle과 뿌리권한을 가진 사용자가 속해있는것을 알수 있다. 여기에 사용자를 추가시켜 같은 집단에 포함시킬수 있다.

【관련명령문】

groupdel : 집단을 삭제한다.

groupmod: 집단의 정보를 변경한다.

groups : 매 사용자가 속한 집단을 보여준다.

groupdel /usr/sbin/groupdel

집단삭제

보안상의 리유로 사용하지 않는 지정한 집단을 제거하는 명령문이다.

형식: groupdel 집단

groupadd 명령으로 추가된 /etc/groups의 집단을 삭제할수 있다.
group명령의 다음에 삭제하려고 하는 집단명만 지정하면 된다.
먼저 groupadd명령으로 grouptest집단을 추가해보자.

```
# groupadd grouptest
```

/etc/group파일에서 grouptest가 있는 행을 grep명령으로 검색출력하도록 한 결과 아래와 같이 grouptest집단이 추가된것을 볼수 있다.

```
kcc@linux:~
# cat /etc/guoup | grep grouptest
grouptest:x:6001:
```

groupdel명령으로 지정한 grouptest집단을 삭제한다.

```
kcc@linux:~
# groupdel grouptest
# cat /etc/group | grep grouptest
```

groupmod

/usr/bin/groupmod

집단정보수정

집단정보를 수정하는 명령문이다.

형식 : groupmod [설정값]집단

-g gid : 지정한 gid에 의해 집단의 ID로 변경한다.
-n name : 지정한 name으로 집단명을 변경한다.
-o : -g설정값과 사용하며 지정한 gid가 존재하여도 중복을 허용한다.

groupmod명령문은 groupadd명령으로 추가된 집단의 이름을 변경하는 경우에 사용하는 명령문이다.
/etc/group파일에 groupadd명령으로 추가한 grouptest집단이 추가되여 있다.

```
kcc@linux:~
# cat /etc/group | grep grouptest
grouptest:x:6001:
```

-n설정값을 사용하여 grouptest집단명을 grouptest1집단명으로 변경할수 있다.

```
kcc@linux:~
# groupmod -n grouptest1 grouptest
# cat /etc/group | grep grouptest
grouptest1:x:6001:
```

이외에도 -g설정값을 지정하여 현재 6001로 되여있는 gid를 변경할수 있다.

```
# groupmod -g 6002 grouptest1
```

groups

/usr/bin/groups

자기의 집단보기

매 사용자가 속한 집단의 이름을 보여주는 명령문이다.

형식 : groups[설정값][사용자]

-help : 도움말을 보여준다.
-version : 판본정보를 보여준다.

groups명령문은 사용자를 지정하지 않으면 현재 체계접속(login)된 사용자가 속한 집단을 보여주며 사용자를 지정하면 지정한 사용자가 속해 있는 집단들을 보여주게 된다.

현재 나의 집단은 admin 이다.

```
kcc@linux:~
$ groups
admin
```

뿌리가 속하나 집단을 한번 확인해보자.

```
kcc@linux:~
$ groups root
root : root bin daemon sys adm disk whell
```

grpck

/usr/sbin/grpck

파일찾기

/etc/group과 /etc/gshadow파일을 검사하는 명령문이다.

형식 : grpck[설정값][파일]

-n : 삭제여부를 확인하지 않는다.

grpck명령은 pwck와 비슷한 명령으로서 /etc/group과 /etc/gshadow의 파일렬을 검사하여 잘못된 부분을 표시하여두고 만일 파일에서 잘못된 부분을 찾으면 삭제할것인가에 대한 여부를 확인한 후 yes/no의 대답을 기다린다. -n 설정값은 모든 질문의 대답을 no로 한다. 일반적으로 /etc/group 파일이 잘못되는 경우는 별로 없다. 집단에 새로운 식별이름을 추가하기 위해 편집하던 도중에 건입력오유가 발생했다고 하자.

아래와 같이 grouptest1집단에 《root》라는 식별이름을 추가하려고 했지만 《roo》만 입력한 경우를 실례로 들수 있다.

```
kcc@linux:~
# cat /etc/group | grep grouptest1
grouptest1:x:6001:roo
```

grpck명령을 실행한 결과 《roo》식별이름정보가 없다고 오유를 내보낸다.

```
kcc@linux:~
# grpck /etc/group
group grouptest1: no user roo
delete member roo'? n
grpck: no changes
```

N을 입력하고 /etc/group파일의 잘못된 부분을 수정할수 있다.

grpconv

/usr/sbin/grpconv

그림자집단생성

/etc/group파일은 /etc/gshadow의 그림자파일로 만드는 명령문이다.

형식 : grpconv

grpconv명령은 pwconv명령과 비슷한 역할로서 /etc/group파일을 /etc/gshadow의 그림자파일을 만들게 된다. 이것을 해제하기 위해서는 grpunconv명령문을 사용한다. 그림자파일(shadow file)에 대한 보다 자세한 설명에 대해서는 pwconv명령을 참고하기 바란다.

참 고

위험한 암호

위험한 암호란 암호해득프로그람을 리용하여 침입자가 쉽게 알아낼수 있는 암호를 말한다.

실례로 /etc/shadow파일을 열어 보았을 때에 암호령역에 7tqNQGFbMOK3s인 식별이름이 있다고 가정하자. 이것은 crypt()함수를 사용하여 원래 우리가 사용하는 암호를 암호화한것이다. crypt()함수는 암호를 암호화할 때에 사용한 salt key를 암호화한 내용의 앞 부분에 붙여넣어 /etc/shadow파일에 보관한다. 그러나 이 함수에는 역 암호화알고리듬이 존재하지 않기때문에 이 암호만으로는 원래 어떤 내용이 들어 있었는가를 알수 없다.

대다수의 암호해득프로그람들에서는 단어파일을 리용해서 빠른 속도로 암호화하여 암호화된 암호와 비교해 나간다. 영어사전에 나오는 단어는 물론이고 사람들이 사용할만한 사람이름, 짧은 문장 등도 포함된 사전을 참조하여 암호해득프로그람은 crypt(《단어》, 《salt key》)의 형식으로 차례로 단어를 대입하여 나간다. 수자의 경우에도 일정한 범위안에서는 간단하게 풀어낼수 있다.

gunzip

/usr/bin/gunzip

압축해제

gzip형식으로 압축된 파일을 해제하는 명령문이다.

형식 : gunzip [설정값][파일]

-c, -stdout, -to-stdout : 표준출력으로 압축파일을 생성하고 원본파일을 그대로 보관한다.

-f, -force : 파일에 다중련결(multilink)이 걸려 있거나 다른 말단에서 이미 사용중인 경우, 또는 파일이 이미 존재하는 등의 여러가지로 문제가 생겨도 압축을 강제로 해제한다.

-h, -help : 도움말을 보여준다.

-l, -list : 압축된 파일의 크기, 압축해제될 때의 크기, 압축률(ratio), 압축이 해제될 때의 파일명 등의 정보를 보여준다.

-L, -license : gunzip의 사용허가를 보여준다.

-r, -recursive : 지정한 등록부를 압축한 경우, 등록부에 속한 파일까지 압축해제를 한다.

-t, -test : 압축된 파일이 제대로 되였는가를 검사한다.

-v, -verbose : 진행상황을 상세하게 보여준다.

-V, -version : 판본정보를 보여준다.

gunzip명령문은 gzip형식으로 압축된 파일을 해제할 때에 사용한다.

gzip -d설정값과 꼭같은 역할을 한다.

먼저 등록부의 파일들을 하나의 파일로 압축해보자. tar cvzf명령은 현재 등록부에 있는 파일을 test.tar파일로 묶은 다음 test.tar파일을 test.tar.gz 파일이름으로 압축한다는 내용이다.

```
kcc@linux:~
# tar cvzf test.tar.gz *
20040417_Command.hwp
20046608.hwp
20040612titlesample1.hwp
20040620.hwp
```

```
G_Command.hwp
G_Command20040707.hwp
Planning.hwp
```

test.tar.gz 파일이름으로 압축된것을 볼수 있다.

```
kcc@linux:~
# ls
20040417_Command.hwp 20040608.hwp 20040612titlesample1.hwp
20040620.hwp     G_Command.hwp     G_Command20040707.hwp
Planning.hwp test.tar.gz
```

.gz확장자는 gunzip명령으로 파일을 압축해제할수 있다.

```
kcc@linux:~
# gunzip test.tar.gz
# ls
20040417_Command.hwp 20040608.hwp 20040612titlesample1.hwp
20040620.hwp     G_Command.hwp     G_Command20040707.hwp
Planning.hwp test.tar
```

혹은 gunzip명령대신에 -d설정값으로도 해제할수 있다.

```
# gzip -d test.tar.gz
```

파일의 압축해제와 tar로 묶여진 파일의 해제를 동시에 하려면 xvzf설정값을 입력해준다.

```
# tar xvzf test.tar.gz
```

gzexe /usr/bin/gzexe

실행파일압축

실행파일을 적절하게 압축하여 용량을 줄이는 명령문이다.

형식 : gzexe[파일...]

gzexe편의프로그람은 실행파일을 압축하는데 압축된 파일이 이전의 파일과 같이 실행가능하게 하여 디스크용량을 줄일수 있는 우점이 있다.

먼저 /bin/cat실행파일을 압축해보자.

```
# gzexe /bin/cat
```

파일을 압축하게 되면 이미 있던 파일이름의 끝에 《~》가 붙여져 자동적으로 예비보관파일을 생성하며 압축된 파일이 이전의 파일로 생성된다.

```
kcc@linux:~
# ls -al /bin/cat*
-rwxr-xr-x   1 root   root    7716  7월 7 02:45 /bin/cat
-rwxr-xr-x   1 root   root    14732 10월 13 2003 /bin/cat~
```

이 압축된 파일은 용량에 있어서 원본의 절반으로 줄었지만 이전과 같이 문제없이 cat의 역할을 할수 있다.

[주의]

압축된 파일은 이전의 파일보다 실행속도가 느린 경향이 있다.

gzip

/usr/bin/gzip

파일압축

gzip방식의 파일압축프로그람으로서 Linux에서 가장 보편적으로 사용되는 압축방식이다.

형식 : gzip[설정값][파일]

gzip명령은 GNU zip로서 압축률이 매우 높은 프로그람이다. 압축에 가장 많이 사용하는 명령문으로 tar명령과 함께 묶어 tar.gz의 확장자로 사용되는 경우도 많다. gzip는 파일압축만을 할수 있고 파일을 묶지는 못한다.

여러개의 파일을 하나의 파일압축으로 묶어 파일전송이나 관리 등의 편리를 위해서는 반드시 tar명령과의 조합이 필요하다.

Gzip는 압축이 성공하는 경우에 기존의 원본파일은 모두 사라지고 .gz의 확장자를 가진 파일이 새롭게 생성되게 된다. 또한 파일의 사용허가와 권한은 기존 파일의것이 그대로 유지된다.

upgrade등록부아래에 있는 모든 파일들을 개개의 .gz파일로 압축해보자.

```
kcc@linux:~

# ls upgrade/
apache-1.3.24-1ch.i686.rpm        openssh-askpass-3.4p1-1ch.i686.rpm
apache-1.3.26-1ch.i686.rpm        openssh-askpass-3.4p1-1ch.i686.rpm
apache-devel-1.3.24-1ch.i686.rpm  openssh-server-3.4p1-1ch.i686.rpm
apache-devel-1.3.26-1ch.i686.rpm  sendmail-server-3.4p1-1ch.i686.rpm
apache-doc-1.3.26-1chi686.rpm     sendmail-cf-8.12.3-2ch.i686.rpm
openssh-3.4p1-1ch.I686.rpm

# gzip -r upgrade
# ls upgrade/
apache-1.3.24-1ch.i686.rpm.gz     openssh-askpass-3.4p1-
1ch.i686.rpm.gz
apache-1.3.26-1ch.i686.rpm.gz     openssh-clients-3.4p1-
1ch.i686.rpm.gz
apache-devel-1.3.24-1ch.i686.rpm.gz     openssh-server-3.4p1-
```

```
apache-devel-1.3.26-1ch.i686.rpm.gz      sendmail-8.12.3-2ch.i686.rpm.gz
apache-doc-1.3.26-1ch.i686.rpm.gz    sendmail-cf-8.12.3-2ch.i686.rpm.gz
openssh-3.4p1-1ch.i686.rpm.gz
```

[주의]

gzip -rd 를 사용해도 같은 결과를 얻을수 있다.

우의 파일들은 개개의 파일로 되여있어 여러가지로 좋지 않다. tar명령과 함께 하나의 파일로 다시 압축해보자.

```
# gunzip -r upgrade
```

upgrade등록부의 모든 파일을 upgrade.tar.gz의 파일이름으로 압축한다.

```
kcc@linux:~
# tar cvzf Uperade.tar.gz uperade/
upgrade/
upgrade/apache-1.3.26-1ch.i686.rpm
upgrade/apache-devel-l.3.26-1ch.i686.rpm
upgrade/apache-doc-1.3.26-1ch.i686.rpm
upgrade/sendmail-8.12.3-2ch.i686.rpm
upgrade/sendmail-cf-8.12.3-2ch.i686.rpm
upgrade/openssh-3.4p1-1ch.i686.rpm
upgrade/openssh-clients-3.4p1-1ch.i686.rpm
ungrade/openssh-askpass-3.4p1-1ch.i686.rpm
upgrade/openssh-server-3.4pl-1ch.i686.rpm
upgrade/apache-1.3.24-1ch.i686.rpm
upgrade/apache-devel-1.3.24-141ch.i686.rpm
```

압축을 풀기 위해서는 다음과 같이 한다.

```
# tar xvzf upgrade.tar.gz
```

H

halt /sbin/halt

체계완료

체계를 안전하게 완료시키는 체계관련명령문이다.

형식 : halt[-n[-w] [-d] [-f]

-n : 완료할 때 동기화(sync)하지 않는다.
-w: 실질적으로 완료하지 않고 /var/log/wtmp기록을 남긴다.
-d : wtmp파일에 기록을 남기지 않는다.
-f : 강제로 완료한다.

halt명령은 /var/log/wtmp 파일에 체계완료기록을 남기고 체계완료나 재기동작업을 한다. 만약 runlevel이 0이나 6이 아닌 상태의 체계에서 halt나 reboot명령이 사용되면 shutdown(8) 명령이 호출된다(-h나 -r의 기능을 수행함).

참 고

체계완료명령문

shutdown, halt, init 등을 리용하여 체계를 완료 혹은 재기동할수 있다.

init 0(체계완료), init 6 (체계의 재시작)

이 명령은 다른 여러 사용자들이 체계에서 작업중인 경우, 작업을 마무리할수 있는 완료경고통보문을 보여주지 않고 곧바로 완료되므로 봉사기체계에서는 사용하지 말아야 한다.

shtdown

이 명령은 체계완료에 대한 경고통보문을 다른 사용자들에게 보여주고 완료하는 시간을 설정하여 매 사용자가 자기의 작업을 마무리할수 있는 시간적여유를 준다.

halt

shutdown -h now와 같은 기능을 하는 명령문이다.

reboot

shutdown -r now와 같은 기능을 하는 명령문이다.

hdparm

/sbin/hdparm

하드디스크정보의 보기 및 설정

하드디스크, CD-ROM 등과 같은 장치의 설정을 확인하고 조정하는 명령문이다.

형식 : hdparm[설정값][장치망]

-a [sectcount] : 파일체계 readahead를 설정하거나 정보를 볼수 있다.
-A [0또는 1] : 하드디스크의 read-lookahead설정을 on/off할수 있다.
-c [chipset_mode] : (E)IDE 32-bit I/O설정을 하거나 정보를 볼수 있다.
-c : IDE의 전원방식상태를 검사한다.
-d [0또는1] : DMA기능을 on/off 할수 있다.
-E : CD-ROM의 속도를 설정한다.
-f : Device exit를 위해 buffer cache를 동기화시킨다.
-g : 하드디스크의 geometry를 볼수 있다
-h : 사용법을 보여준다.
-i : 기동시간에 얻어지는 구동기의 정보를 볼수 있다.
-l : 구동기로부터 직접 구동기의 정보를 볼수 있다.
-k [0 또는 1]: keep_settings_over_rest 를 설정하거나 정보를 볼수 있다.
-K [0 또는 1] : 구동기의 keep_features_over_reset를 설정한다.
-L [0 또는 1] : 구동기의 자물쇠(door lock)를 설정한다.
-m [sectcount]: 하드디스크의 다중분구수(multi-sector number)를 설정하거나 정보를 볼수 있다.
-p [0~5중 하나j] : 하드디스크의 prefetch수를 설정한다.
-P [sectcount]: 하드디스크의 내부적인 prefetch 구조를 위한 최대분구수를 설정한다.
-q : 설정값을 화면에 추력하지 않는다. -I, -v, -t, -T 등의 설정값들에는 적용되지 않는다.
-r [0 또는1]: 읽기전용(read-only)방식으로 설정하거나 정보를 보여준다.

-R : IDE대면부를 등록한다.
-S [timeout] : 구동기의 standby(spindown)시간을 설정한다.
-T : csche읽기시간을 볼수 있다.
-t : 장치읽기시간을 볼수 있다.
-u [0 또는 1] : 장치의 interrupt-unmaask를 설정하거나 정보를 보여준다.
-U : IDE대면부를 제거한다.
-v : -i 설정값을 제외한 모든 설정을 보여준다.
-W [0 또는 1] : IDE구동기의 write-cachig기능을 열거나 닫는다.
-X [xfermode] : 새로운 (E)IDE/ATA2구동기를 위해 IDE전송방식을 설정한다.
-y : IDE구동기를 standby방식으로 전환한다.
-Y : IDE구동기를 sleep방식으로 전환한다.
-Z : seagate auto-powersaving 방식을 무효화(disable)시킨다.

하드디스크의 정보를 보도록 하자. 아래의 출력표시를 통해 32-bit I/O를 지원하고 DMA기능을 사용하고있다는것을 알수 있다.

```
kcc@linux:~

# hdparm /dev/hda

/dev/hda:
multcount     =16(on)
I/O support   =1(32-bit)
Unmaskirq     =1 (on)
using_dma     =1 (on)
keepsettigs   =0 (off)
nowerr        =0 (off)
readonly      =0 (off)
readahead     =8 (on)
geometry      =1826/255/63, sectors=29336832, start=0
```

CD-ROM장치의 DMA기능을 보도록 하자. 이 기능은 using-dma로서 설정이 on으로 되여있다.

```
kcc@linux:~
# hdparm -d /dev/hdd

/dev/hdd:
using_dma = 1 (on)
```

DMA기능을 off로 설정하려면 -d 0설정값을 사용한다.

```
kcc@linux:~
#  hdparm -d 0 /dev/hdd

/dev/hdd:
setting using_dma to 0 (off)
using_dma = 0 (off)
```

일부 RedHat계렬의 Linux배포판이 설치된 체계에서 LG CD-ROM 40배속과 48배속을 사용하는 경우에 CD가 인식되지 않거나 정지현상이 발생하는 경우가 있다. 이것은 체계의 DMA기능사용에서 발생되는 문제인데 hdparm명령문으로 DMA사용을 금지(off)시키면 해결된다.

```
# hdparm -c1 d0/dev/cdrom
```

참 고

사용하지 않을 때 하드디스크 정지하기

5분이상 휴식(idle)상태로 되는 경우에 하드디스크가 자동적으로 전원정지(power down)상태로 되도록 만들수 있다. 다음과 같이 입력해보자.

```
# hdparm -S 60 /dev/hda
```

head

/usr/bin/head

파일앞부분의 보기

파일의 앞부분부터 출력시키려는 행을 지정하여 출력표시하도록 하는 명령문이다.

형식 : head [설정값][파일이름]

-행수 : 출력하도록 하려는 행수를 지정하여준다. 기본적으로 10을 지정하여주어 10개의 행을 출력하도록 한다.

-c, -bytes=SIZE : 출력하도록 하려는 용량을 지정할수 있다.

SIZE는 b(block=512byte), k(Kbyte), m(Mbyte) 등을 수자뒤에 붙여 용량단위를 구분하여주어야 한다. 뒤에 이 단위가 없는 경우에 byte단위로 보여준다.

-n, -lins=행수 : 출력하도록 하려는 행수를 지정하여준다.

-q, -quiet, -silent : 출력할 때에 파일명을 보여주지 않는다.

-v, -verbose : 출력하는 파일명을 보여준다.

—help : 도움말을 보여준다.

—version : 판본을 보여준다.

파일의 앞부분을 보여주는 명령문이다. 보려고 하는 부분을 행수, 용량으로 지정해줄수 있다. 이 명령은 주로 긴 파일의 앞부분을 보려고 할 때 많이 사용되는것으로서 리력파일을 확인할 때에도 tail명령과 함께 많이 사용된다.

head명령문을 리용하여 mail파일의 처음 10행을 읽어보자. -v 설정값을 사용하면 출력되는 파일이름을 먼저 보여준다. 이 설정값은 출력내용을 파일로 보관할 때에 이 보관내용이 어느 파일의 앞부분인가를 보여주도록 표시하는데 사용할수 있다.

```
kcc@linux:~

# head -v /var/mail/ebook
==> /var/spool/mail/root <==
From root  wed Feb 20 04:03:43 2004
```

```
Return-Path:<root@kcctongbo.com>
Received: (from root@localhost)
by kcctongbo.com (8.10.1/8.10.1) id g1JJ3gK00356
for root: Wed, 20 Feb 2004 04:03:42 +0900
Date: Wed, 20 Feb 2004 04:03:42 +0900
From:<root root@kcctongbo.com>
Message-Id:<200402191903.g1JJ3gK00356@kcctongbo.com>
To: root@dizikarma.com
Subjecy: File integrity report
```

우에서 10개행으로 이루어진 우편의 서두부분을 출력해보았다.

그러면 이번에는 12번째행까지 출력하도록 하자. 다음과 같이 출력시킬 범위를 지정하여준다.

```
kcc@linux:~

# head -12 /var/mail/ebook
==> /var/spool/mail/root <==
From root  Wed Feb 20 04:03:43 2004
Return-Path:<root@kcctongbo.com>
Received: (from root@localhost)
          by kcctongbo.com (8.10.1./8.10.1) id g1JJ3gK00356
          for root: Wed, 20 Feb 2004 04:03:422 +0900
Date: Wed, 20 Feb 2004 04:03:42 +0900
From: root<root@kcctongbo.com>
Message-id:<200402191903.g1JJ3gK00356@kcctongbo.com>
To: root@kcctongbo.com
Subject: File integrity report
Hello~~!!!!
You must read this mail and send your answer.
```

파일용량을 지정하여 일정한 용량까지만을 출력하도록 하는 방법도 있는데 이때에는 -c설정값을 사용한다.

```
Kcc bit@linux:~

#  head -v -c 74 /var/spool/mail/root
==> var/spool/mail/root  <==
From root  Wed Feb 20  4:03:43 2004
Return-Path:<root@office.kcccom.com>
```

우와 같이 명령을 입력하면 74bytes의 용량만큼 출력한다.

【관련명령문】

cat : 파일을 첫행부터 아래방향으로 보여준다.

tai : 파일의 마지막행을 보여준다.

tac : 파일을 마지막행부터 우의 방향으로 보여준다.

host

/usr/bin/host

호스트정보보기

령역의 IP나 IP에 해당한 령역을 알려고 할 때에 사용하는 명령문이다.

형식 : host[설정값][domain,IP][봉사기]

봉사기: 령역(domain)이나 IP를 검색할 이름봉사기를 지정한다. 지정하지 않는 경우 체계에 등록된 령역봉사기를 검색한다(/etc/resolv.conf).

-I zone : zone아래의 모든 체계를 출력한다.

-v : 자세한 정보를 보여준다.

-w : DNS봉사기응답을 기다린다.

-r : 반복처리하지 않는다.

-d : 오유수정방식으로 보여준다.

-t [type] : type를 지정하여 정보를 얻는다(A : 호스트 IP주소, NS : 검색한 호스트의 nameserver호스트이름, PTR : domain name pointer, ANY : type의 모든 정보)

-a : 《-t ANY》와 같은 기능을 한다.

호스트이름은 알고있지만 IP주소를 모르는 경우, 혹은 그 반대의 경우에 사용한다. host를 리용하면 다만 IP주소뿐만아니라 해당한 호스트명(host name)이 가지고있는 아래준위호스트명도 검색할수 있다. host는 체계에 등록되여있는 DNS봉사기를 검색한다. 만약 다른 DNS봉사기를 리용하려는 경우에는 검색하려는 호스트명/IP주소뒤에 봉사기의 주소를 써준다.

www.kcc.go.kp의 IP주소를 검색하며 이 주소의 검색에 kccnet의 DNS를 리용하는 경우를 실례로 들어보자.

사용하는 DNS봉사기의 정보와 검색한 IP주소가 출력된다.

```
kcc@linux:~
# host www.kcc.co.kp ktb.kccnet.net
Using domain server:
Name: ktb.kccnet.net
Address: 168.126.63.1
Aliases:

Rc:kcc.go.kp has address 211.32.119.151
Rc:kcc.go.kp has address 211.32.119.151
```

검색하려는 type를 -t설정값으로 정해주면 kcc.go.kp의 이름봉사기의 령역을 알수 있다.

```
kcc@linux:~
# host -t NS kcc.go.kp
kcc.go.kp.name server ns0.kcc.go.kp
kcc.go.kp.name.server ns2.kcc.go.kp
```

참 고

DNS(Domain Name System)봉사기

IP주소로 이루어진 수자체계의 인터네트주소는 일반 사람들이 리해하기 힘든것으로 하여 쉽게 기억되고 사용될수 있는 령역이름이 필요하였다. 물론 매 호스트체계는 IP주소로 찾아가야 하므로 IP주소와 령역이름을 중계할수 있는 체계가 필요하게 되였다. 이와 같이 호스트이름과 IP주소를 호상 변환하며 중계역할을 하는것을 DNS봉사기라고 한다. DNS봉사기는 분산형자료기지로 현재 DNS봉사기에서 주소를 찾지 못하는 경우에 웃준위의 DNS봉사기에 접속하여 찾게 된다. DNS는 망우의 콤퓨터와 콤퓨터를 련결하여 쉽게 통신하기 위한 도구인 동시에 광범한 자료기지라고 할수 있다.

hostid

/usr/bin/hosted

호스트 ID지정

호스트 ID를 지정하거나 보여주는 명령문이다.

형식 : hostid[설정값]

-help : 간단한 도움말을 보여준다.
-version : 판본을 보여준다.

hostid명령은 현재 호스트 ID 번호를 10진수로 보여준다. 이때 -v설정값이 주어지면 이 번호를 10진수와 16진수의 두가지로 보여준다. 이것은 그 호스트의 고유번호이며 다른 호스트들과 구별할 때, 그리고 인터네트주소를 부여할 때에 사용된다.

```
kcc@linux:~
# hostid
283d70e9
```

이 값은 체계관리자가 명령행인수를 사용하여 새롭게 지정할수 있는데 이 값은 /etc/hostid 파일에 보관되고 필요한 경우에 읽혀진다.

참 고

호스트 ID와 망 ID

호스트 ID와 망 ID는 모두 IP를 구성하는 요소이다.

망 ID 또는 망주소는 IP경로기에 의해 묶여져있는 같은 물리적망에 존재한다는 것을 의미한다. 다시말해서 같은 망에 묶여져있는 체계는 같은 망 ID를 사용한다는 것이다. 또한 인터네트에서 이 망 ID는 반드시 유일해야 한다.

호스트 ID(혹은 호스트주소)는 망내에서 전용작업기(Workstation), 봉사기, 경로기, 기타 TCP/IP호스트를 구분하는것이다. 매 호스트의 주소는 반드시 망 ID에 대해 유효해야 하며 같은 망안에서 유일해야 한다.

hostname

/bin/hostname

체계이름보기

체계이름을 확인하고 설정하는 명령이다.

형식 : hostname [설정값] 파일이름

-a, -alias : alias명을 보여준다.

-d, -domain : 령역을 보여준다.

-F, -file **파일명** : 파일로부터 호스트이름을 받아서 설정한다.

-f, -fqdn, -long

-h, -help : 도움말을 보여준다.

-i, -ip : 호스트의 IP주소를 보여준다.

-n, -node : Display the DECnet node name

-s, -short : 짧은 호스트이름을 보여준다. FQDN에서 앞의 첫번째 《.》전의 부분을 보여준다.

-v, -version : 판본을 보여준다.

-v, -verbose : 호스트설정이나 호스트이름을 자세히 보여준다.

-y, -yp, -nis : NIS령역이름을 보여준다. 또한 파일을 받아와 NIS령역이름을 설정할수 있다.

체계이름을 보여주며 변경할수도 있다. 설정값이 없는 기본형으로 사용하면 FQDN형태로 출력된다.

```
kcc@linux:~
# hostname -v
gethostname()=office.kcccom.com'
www.kcctongbo.com
```

호스트이름을 변경한다.

```
kcc@linux:~
# hostname -v ftp.kcctongbo.com
Seiting hostname to ftp.kcctongbo.com
```

참 고

FQDN(Fully Qualified Domain Name)

Linux설치시에 망설정과정에서 령역을 입력하는 창이 나오는것을 보았을것이다. 령역을 입력하려고 할 때에 www.kcctongbo.co.kp가 령역인지, kcctongbo.co.kp가 령역인지 명확치 못해하는 경우가 있는데 이와 같은 문제때문에 FQDN이라는 단어를 사용한다. 사실 호스트이름은 앞의 www이고 령역은 kcctongbo.com이다. 그리고 www.kcctongbo.co.kp는 FQDN이라는 용어로 정의한다. 이것은 령역과 호스트를 함께 묶어 사용하는것을 의미한다.

www	kcctongbo.com	www.kcctongbo.com
호스트명	령역	FQDN

IPv6

현재 사용되는 TCP/IP주소의 구조는 32bit방식의 IPv4로서 인터네트사용의 폭발적인 증가와 함께 현재 그 수가 부족해지는 상태에 이르렀으며 이로부터 나온것이 IPv6이다. IPv6은 128bit의 주소구조를 가진 다음세대의 TCP/IP표준이다. IPv6은 32bit의 주소정보를 가지는 IPv4보다 4배나 많은 정보를 가질수 있으므로 주소할당요구가 증가할수록 관리가 복잡하고 비효률적으로 되는 한계를 극복할수 있다. IPv6은 그의 모든 규격내용이 인터네트표준을 정의하는 IETF(Internet Engineering Task Force)의 권고대로 설계되여 업계표준으로 도입실용화되고있다.

I

id /usr/bin/id

식별이름 ID의 확인

사용자의 UID, GID 번호 및 현재 쉘에서의 ID정보를 보여준다.

형식 : id[설정값] [사용자명]

-g, -group : 사용자의 집단 ID만을 보여준다.
-G, -groups : 추가 집단의 ID만을 보여준다.
-help : 사용법을 보여준다.
-n, -name : -u, -g, -G설정값과 같이 쓰이여 해당하는 ID의 이름만을 보여준다.
-r, -real : -u. -g, -G설정값과 같이 쓰이여 해당하는 실제 ID를 보여준다.
-version : 판본정보를 보여준다.

id명령문은 현재 사용자의 실지 ID와 유효사용자 ID, 집단 ID를 보여준다.

내부 bash변수인 $UID, $EUID $GROUPS와 짝을 이룬다. 만일 사용자를 지정하면 해당하는 사용자의 정보를 보여주게 된다.

id명령으로 현재 망접속한 admin 식별이름의 정보를 보기로 하자.

[주의]
EUID에 대한 설명은 chmod부분을 참고할것.

```
kcc@linux:~

$ id
uid=500(admin)  gid=500(admin)  groups=500(admin),22(cdrom)
80(cdwriter),81(audio)
```

우의 화면출력내용을 통해 망접속한 사용자는 admin이 식별이름정보이고 uid는 500, gid가 500이라는것을 알수 있다. 또한 admin식별이름은 gid 22, 80, 81의 집단에도 속한 사용자라는것을 알수있다.

echo $UID명령으로 현재 admin식별이름의 uid를 알아볼수도 있다.

```
kcc@linux:~
$ echo $UID
500
```

ifconfig

/sbin/ifconfig

망대면부설정

망대면부와 망환경을 설정하는 명령문이다.

형식 : ifconfig [interface]
ifconfig interface[type]설정값|주소

interface : 망대면부이름이다. 대체로 NIC가 설정된 경우에는 eth0, eth1로 되고 PPP로 련결된 경우에는 ppp0, ppp1의 이름으로 된다.

type : 지정한 망대면부에서 사용할 통신규약을 지정한다. 현재 지원되는 것으로서는 inet(TCP/IP를 사용하는 경우), inet6(IPv6), ax25(AMPR Packtet Radio), d(Appletalk Phase 2), ipx(Novell IPX) 등이 있다.

up : 지정한 망대면부를 활성화한다.

down : 지정한 망대면부를 비활성화한다.

[-]arp : ARP통신규약을 활성화/비활성화한다.

[-]promisc : promiscuous방식을 활성화/비활성화한다. promiscuous방식을 활성화하면 망대면부를 통과하는 모든 파케트를 받는다.

[-]allmulti : all-multicast방식을 활성화/비활성화한다. all-multicast방식을 활성화화면 대면부를 통과하는 모든 파케트를 받는다.

metric N : 망대면부 metric을 설정한다.

mtu N : 망대면부 MTU를 설정한다.

dstaddr addr : PPP원격 IP주소를 설정한다.

netmask addr : 망대면부의 netmask를 설정한다.

add addr/prefixlen : 망대면부에 IPv6주소를 부여한다.

del addr/prefixlen : 망대면부의 IPv6주소를 제거한다.

irq addr : 장치의 IRQ 주소를 지정한다.

io_addr addr : 망대면부장치의 IO주소를 지정한다.

mem_start addr : 망대면부장치의 공유주기억의 시작주소를 지정한다.

media type : 망대면부장치의 물리적형식을 설정한다. 물리적형식에는 10base2(thin Ethernet), 10baseT(twisted-pair 10Mbps Ethernet), AUI(external transceiver) 등이 있다.

[-]broadcast [addr] : 망대면부의 방송(broadcast)주소를 설정한다.
[-]pointopoint [addr] : 망대면부의 점대점(point-to-point)방식을 활성화하고 주소를 설정한다.
hw claas address : 망대면부의 하드웨어주소를 설정한다.
multicast : 망대면부를 다중방송(multicast)방식으로 설정한다.
address : 망대면부에 IP주소를 설정한다.
txqueuelen length : 망대면부장치의 전송대기렬길이를 설정한다.

ifconfig는 보통 망대면부의 설정을 변경하기 위해 사용한다. 먼저 -a설정값을 사용하여 설정되어있는 망대면부를 확인해보자.

```
kcc@linux:~

# ifconfig -a
eth0     Link encap:Ethernet  Hwaddr 00:50:BF:28:2A:4A
         inet    addr:.211.25.253.59         Bcast:211.255.253.63
Mask:255.255.255.224
         UP BROADCST RUNNING MULTICAST MTU:1500 Metric :1
         RX packets:887134 errors:0 dropped:0 overruns:0 frame:0
         TX packets:803985 erros:0 dropped:0 overruns:0 carrier:0
         collisions:0
         RX bytes:672339450 (641.1 Mb) TX bytes:111521026 (106.3 Mb)

lo       Link encap:Local Loopback
         inet addr:l27.0.0.1   Mask:255.0.0.0
         UP LOOPPBACK RUNNING MTU:16436 Metric:1
         RX packets:7149 errors:0 dropped:0 overruns:0 frame:0
         TX packets:7149 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0
         RX bytes:3043077 (2.9 Mb) TX bytes:3043077 (2.9 Mb)
```

IP설정

eth1망대면부에 아래와 같이 망의 IP주소와 netmask를 설정해보자.

```
# ifconfig eth1 192.168.0.1 netmask 255.255.255.0
```

설정된 eth1의 망대면부설정을 확인해보자.

```
kcc@linux:~

# iftconfig eth1
eth1   Link encap:Ethernet HWaddr 00:50:BF:15:DB:41
       inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:58429 errors:0 dropped:0 overruns:0 frame:0
       TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0
       RX byies:5602420 (5.3 Mb) Tx bytes:6000 (5.8 kb)
```

망대면부를 활성화하기

eth1망대면부를 활성화한다.

```
# ifconfig eth1 up
```

아래와 같이 ifup명령으로도 활성화할수 있다.

```
# ifup ethl
```

[주의]

ifdown eth1이라고 입력하는 경우에도 같은 결과를 얻을수 있다.

망대면부를 비활성화하기

eth1망대면부를 비활성화하기 위해서는 다음과 같이 입력한다.

```
# ifconfig eth1down
```

기본게트웨이의 설정

기본게트웨이를 211.255.253.62로 설정해보자.

```
kcc@linux:~

#  route add -net detault gw 211.255.253.62
```

설정된 게트웨이는 route명령으로 확인할수 있다.

```
kcc@linux:~

# route
Kernel IP routing table
Destination     Gateway  Genmask          Flags   Metric  Ref  Use  Iface
211.255.253.32 *     255.255.255.224  U       0       0    0    eth0
192.168.0.0     *     255.255.255.0    U       0       0    0    eth1
127.0.0.0       *     255.0.0.0        U       0       0    0    lo
default         211.255.253.62         0.0.0.0 UG      0    0    0 eth0
```

참　　고

IP alias를 설정하기

eth0망대면부의 alias인 eth0:0을 설정해보자.

```
kcc@linux:~

# ifconfig eth0:0 192.168.1.1 netmask 255.255.255.0
# ifconfig eth0:0
eth0:0  Link encap:E:Ethernet   HWaddr 00:50:BF:28:2A:4A
        inet addr:192.168.1.1  Bcast:192.168.1.255 Mask:255.255.255.0
        UP BROADCAST RUNING MULTICAST MTU:1500   Metric:1
```

망대면부 PROMISC방식의 설정 및 확인

promisc방식은 그의 어휘적의미와 같이 국부망에서 오가는 파케트정보들을 볼수 있게 한다. 이것은 내부망의 파케트들을 부당한 목적으로 훔쳐볼수도 있게 하므로 이것을 검사확인해볼 필요가 있다.

```
kcc@linux:~

# ifconfig eth1 promisc
# ifconfig eth1
eth1   Link encap:Ethernet Hwaddr 00:50:BF:15:DB:41
       inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
       UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
       RXpackets:58687 errors:0 dropped:0 overruns:0 frame:0
       TX packets:100 errors:0 dropped: overruns:0 carrier:0
       collisions:0
       RX byles:5626647(5.3 Mb) TX bytes:6000 (5.8Kb)
```

promisc방식을 아래와 같이 해제한다.

```
kcc@linux:~

# ifconfig eth1 -promisc
# ifconfig eth1
eth1   Link encap:Ethernet Hwaddr 00:50:BF:15:DB:41
       inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:58761 errors:0 dropped:0 overruns:0 frame:0
       TX packets:100 errors:0 dropped: overruns:0 carrier:0
       collisions:0
       RX byles:5633361 (5.3 Mb) TX bytes:6000 (5.8Kb)
```

info

/usr/bin/info

하이퍼본문형의 문서페지

일반적인 체계설명에 하이퍼본문적인 기능이 추가된 문서페지(manual page).

형식 : info[설정값]항목

-directory 경로이름 : 경로이름을 추가하여 info문서를 찾을수 있는 등록부경로가 추가된다. 기본값은 /usr/info 혹은 /usr/local/info등록부이다.

-f file, -file file : 지정한 file로 info파일을 보여준다.

-n node, -node node : 지정한 node로 마디의 파일을 보여준다.

-o file, -output file : 화면으로 결과를 보여주지 않고 지정한 file로 결과를 복사한다.

-help : 도움말을 보여준다.

-version : 판본정보를 보여준다.

info는 조작체계 및 련관된 프로그람들에 대한 문서를 제공한다. info파일은 대부분 /usr/local/info 혹은 /usr/info등록부에 포함되여있다.

다른 하나의 대표적인것으로서 man명령문으로 문서페지(manual page)를 참고하는 명령문이 있다.

info가 실행되면 다음과 같은 명령을 사용할수 있다.

H	info에 대한 리용법을 호출한다.
?	Info에 대한 명령문요약을 보여준다.
Ctrl-g	작업진행중지, 명령취소
Ctrl-l	화면을 새롭게 갱신한다.

다른 마디(node)에로 이동하기

n	다음의 마디에로(next)
p	이전의 마디에로(previous)

u	우의 마디에로(up)
m	특정한 마디를 지정하여 지정한 마디에로 이동한다. 이때 정확한 마디이름을 입력할 필요는 없다.
f	호상참조사용으로서 f를 입력하면 호상참조할 이름을 물어본다. 이때 m의 사용법과 같이 사용하면 된다.
l	마지막마디에로 이동한다.

마디안에서의 이동하기

Space	한 화면 아래로(다음으로, 앞으로) 이동
DEL	한 화면 우로(이전으로, 뒤로) 이동
b	마디의 처음으로

고급명령

q	Info를 완료한다.
1	마디안에 있는 안내문의 첫번째 항목으로 이동한다. 수자를 지정하게 되면 지정한 항목으로 이동한다.
g	지정한 특정마디에로 이동한다.
s	문자렬을 찾기 위해 화면의 아래부분에 문자렬입력창을 현시한다. 현재 마디에서 지정한 문자렬을 찾아 발견되면 그곳으로 유표를 이동한다.

info명령은 이전에 man명령보다 적게 쓰이였지만 그 편리성으로 하여 점차 널리 사용되고있다.

init /sbin/init

초기화프로세스조종

기동시에 실행준위에 따라 여러 프로세스를 호출하여 시작한다.

형식 : /sbin/init [-t sec][0123456SsQq]

0,1,2,3,4,5,6 : 매개의 실행준위로 다시 시작한다. 매 준위는 /etc/inittab에서 확인할수 있다.

Q or q : /sbin/init가 /etc/inittab파일을 다시 읽는다.

S or s : /sbin/init에서 단일사용자방식으로 전환한다.

-t sec : 지정한 sec초만큼 init가 TERM signal을 보내고 몇초를 기다린 다음에 KILL signal을 보낸다. 초기값은 5초이다.

[주의]

fork란 프로세스가 자기자신을 복제하는것을 말한다.

핵심부(kernel)가 주기억기에 적재되면 뒤이어 init명령이 실행된다.

init는 자신을 fork하여 여러 프로세스를 시작하고 프로세스를 시작하는 스크립트를 호출한다. 우선 /etc/iniitab를 읽어 체계의 실행준위(runlevel)를 결정하고 망접속을 위한 getty프로그람과 여러가지 데몬을 실행시킨다.

실행준위는 Linux배포판마다 다를수 있는데 RedHat에서는 아래의 화면출력례에서 보여주는바와 같다. initdefault로 실행준위를 5로 지정하였는데 5는 X11로 실행되면 된다. 만일 조종탁방식(console mode)으로 기동하려는 경우에는 id:5:initdefault를 id:3:initdefault로 변경하면 된다.

#Run gettys in standard runlevels주해아래의 mingetty는 망접속을 관리하는 getty류형의 명령문이다. 실행준위 5로 기동한후에 [Ctrl+Alt+F1]부터 [F6]건을 입력하면 매개의 조종탁체계접속(console login)화면을 볼수 있는것은 이러한 의미이다. 만일 조종탁화면을 줄이려고 하는 경우에는 원하는 수자만을 남겨두고 나머지는 모두 삭제하거나 앞줄에 주해(#)처리한다.

inittab파일을 수정한 다음에는 init -q명령으로 수정된 내용을 적용할수 있다. inittab파일의 맨 마지막행의 x:5:respawn:/etc/X11/prefdm -nodaemon은 실행준위 5에서의 X Window망접속프로그람을 실행하는것이다.

```
kcc@linux:~

# cat /etc/inittab
-----------------------------------------------------------------
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
Ud::once:/sbin/uqdate

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
# When our UPS tells us power has failed, assume we have a few minutes
# of power left, Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
Pf::powerfail:/sbin/shudown -f -h +2 "Power Failure: System Shutting Down"
```

```
# if power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Run gettys in standard runlevels
1:2345:respawn:sbin/mingetty tty1
2:2345:respawn:sbin/mingetty tty2
3:2345:respawn:sbin/mingetty tty3
4:2345:respawn:sbin/mingetty tty4
5:2345:respawn:sbin/mingetty tty5
6:2345:respawn:sbin/mingetty tty6

# Run xdm in runlevel 5
# xdm is now a separate service
x:5:respawn:/etc/x11/prefdm   nodaemon
```

insmod

/sbin/insmod

핵심부모듈을 불러들이기

사용할수 있는 핵심부모듈(kernel module)을 적재하여 동작하게 하는 명령문이다.

형식 : insmod [설정값]모듈파일[symbol=value]

-f : 핵심부판본이 달라도 강제로 모듈을 적재한다.
-h : 사용법을 보여준다.
-k : 모듈을 자동으로 삭제가능하도록 한다.
-L : 동일한 모듈이 동시에 적재되지 못하게 한다.
-m : 적재배치표(load map: 주기억기에 상주하고있는 프로그람이나 자료의 전체 혹은 선택된 부분들의 기억위치 또는 크기를 나타내는 목록)를 생성한다.
-n : 모듈을 적재하지 않고 보여주기만 한다.
-p : 모듈이 핵심부와 부합되는가를 확인한다.
-s : syslog에 오유를 출력한다.
-v : 오유를 상세히 보여준다.
-V : 판본을 보여준다.
-o NAME : 모듈을 지정한 NAME으로 설정한다.
-p prefix : 핵심부나 모듈을 prefix 한다.

insmod명령은 핵심부의 모듈을 적재할 때 사용하는 명령문이다.

insmod와 함께 보다 지능적인 modprobe명령이 있다. 모듈을 적재할수 있는 모듈등록부는 RedHat의 경우, /lib/modules/핵심부판본/kernel/에 있다. /lib/modules등록부 다음의 등록부는 매 핵심부판본을 따른다.

매 아래준위등록부에서 .o으로 끝나는 모듈파일을 확인해볼수 있다.

```
kcc@linux:~
# cd/lib/modules/2.4.13-1h1/kernel/
# ls
arch  drivers  fs  net
```

lsmod명령문으로 현재 주기억기에 적재된 모듈들을 볼수 있다.

ip_tables모듈, 음향관련모듈, 이써네트기판모듈, ext3모듈, usb관련모듈등이 있다.

```
kcc@linux:~

# ismod
Mdule        Size   Used by    Tainted: P
ip_tables    11200        0         (unused)
ymfpci       42660        0         (autoclean)
uart401      6560         0         (autoclean)   [ymfpci]
sound        59052        0         (autoclean)   [uart401]
ac97_codec   9504         0         (autoclean)   [ymfpci]
soundcore    4324         4         (autoclean)   [ymfpci sound]
8139too      17440  2
ext3         61600  2     (autoclean)
jbd          40452        2         (autoclean)   [ext3]
usb-uhci     21764        0         [unused]
usbcore      51744        1         [usb-uhci]
```

우의 모듈가운데서 ip_tables모듈을 rmmod명령으로 주기억기에서 제거해보자. 제거된 ip_tables 모듈은 modprove명령이나 insmod명령으로 다시 적재할수 있다.

```
# rmmod ip_tables
```

```
kcc@linux:~

# insmod ip_tables
Using /lib/moduyles/2.4.13-1hl/kernet/net/ipv4/netfilter/ip_tables.o
# lsmod
Module       Size     Used by  Tainted:        P
ip_tables    11200    0        (unsed)
ymfpci       42660    0        (autoclean)
uart401      6560     0        (autoclean)     [ymfpci]
sound        59052    0        (autoclean)     [uary401]
ac97_codec   9504     0        (autoclean)     [ymfpci]
```

```
soundcore   4324    4       (autoclean)     [ymfpci sound]
autofs      10948   0       (autoclean)     (unused)
8139too     17440   2
ipx         16404   0       (autoclean)
ext3        61600   2       (autoclean)
jbd         40452   2       (autoclean)     [ext3]
usb-uhci    21764   0       (unused)
usbcore     51744   1       [usb-uhci]
```

이와 같이 insmod, rmmod, lsmod 등의 명령으로 자유롭게 모듈들을 관리할수 있다.

【관련명령문】

depmod : 핵심부모듈을 적재하기 위한 의존성을 다루는 명령문

modprobe : 핵심부모듈을 적재하는 명령문

참 고

핵심부웨브봉사기(httpd)를 사용하기

핵심부웨브봉사기가속기데몬(kHTTPd)은 핵심부에 내장된 웨브봉사기로서 모듈의 형태로 사용할수 있다. 파일체계의 파일만을 봉사하기때문에 cgi와 같은 동적내용물(dynamic contents)은 다룰수 없다. 그러나 파일만을 사용할 때는 Apache와 같은 일반 웨브봉사기보다 빠르게 봉사할수 있는 우점이 있으므로 다른 웨브봉사기와 함께 사용하면 매우 효률적이다. 즉 일반파일봉사는 kHTTPd가 처리하고 cgi는 Apache가 처리하도록 하면 좋은 효과를 낼수 있다. 이 프로그람은 아직 완벽하지 못하며 웨브봉사기로 사용하지 않는 봉사기에서 동작하는 경우, 보안상의 문제점을 발생시킬수 있으므로 모듈로 선택하게 되여있다. 핵심부를 번역할 때에 해당한 설정값을 입력하여 모듈로 번역한다.

install

/usr/bin/install

파일설치

지정한 속성대로 정해진 위치에 파일을 복사하는 명령문이다.

형식 : install[설정값][파일]등록부

-d -directory : 대상등록부를 지정하는데 만일 경로가 없을 때에는 등록부를 만든다.
-g group, -group group : 새로운 파일의 집단으로 지정한 group ID나 group이름으로 설정한다.
-m mode, -mode mode : 새로운 파일의 권한을 지정한 mode값으로 사용한다. 기본값은 0755이다.
-o [owner], -owner [owner] : 지정한 owner로 소유자를 지정한다. 만일 지정하지 않으면 root로 실행시에 지정하는 소유자로 변경한다.
-s, -strip : symbol table을 제거한다.
-help : 도움말을 보여준다.
-version : 판본정보를 보여준다.

install명령은 파일의 권한속성들과 소유자, 집단을 지정하여 복사한다.

cp명령과 비슷하지만 필요에 따라 파일이 복사될 경로를 만들고 다른 소유자나 집단을 지정하는것과 같은 다양한 기능을 포함하고있으므로 번역된 프로그람을 정확한 위치에 설치하는데 많이 사용된다. 그러므로 Makefile에서 사용된다. 현재 등록부의 file_change_or_mv.sh 스크립트실행파일을 /bin 등록부아래에 설치해보자.

```
# install -o admin -g admin file_change_or_mv.sh /bin/
```

설치된 file_change_or_mv.sh파일은 whereis명령에 의해 위치를 찾을 수 있으며 -o설정값으로 지정한 소유자와 -g설정값으로 지정한 집단으로 설치되게 된다.

ipchains /sbin/ipchains

Kernel 2.2.x판본의 방화벽관리도구

Linux Kernel 2.2.x판본에서 방화벽, 규칙설정, IP masquerading을 조정하는 핵심부도구이다.

형식 : ipchains 명령[설정값]

-A chain, -append chain : chain을 추가한다.

-C chain, -check chain : 설정한 chain방책을 검사한다.

-D chain, -delete chain : chain에서 규칙을 삭제한다.

-D chain 규칙번호, -delete chain 규칙번호 : chain방책(chain policy)내에서 지정한 규칙번호(rule number)를 삭제한다. 만일 규칙번호가 1이라면 chain규칙의 첫번째 규칙을 삭제한다.

-F [chain] -flush [chain] : 모든 chain방책 또는 지정한 chain방책을 삭제한다.

-l chain [규칙번호], -insert chain 규칙번호 : chain방책에 지정한 수자번째에 삽입하거나 마지막에 규칙을 삽입한다.

-L [chain], -list [chins] : 모든 chain방책 또는 지정한 chain방책을 본다.

-ML, -masquerading -list : 현재 masquerading접속된 상황을 본다.

-MS tcp tcpfin udp, -masquerading -set tcp tcpfin udp : masquerading 시간초과(time out)값을 설정한다.

-N chain, -new-chain chain : 새로운 chain방책을 만든다.

-P chain target, -policy chain target : chain방책을 지정한 chain 방책으로 교체한다.

-R chain 규칙번호, -replace chain 규칙번호 : chain방책내의 지정한 수자번째의 규칙을 교체한다.

-Z [chain1 -zero [chain] : 모든 chain방책 또는 지정한 chain방책을 령(zero)으로 만든다.

-b, —bidirectional : 두가지 규칙을 삽입한다(-s와 -d).

-p [!] proto, —protocol [!] proto : 통신규약을 지정한다. !값은 제외의 의미이다.

-s [!] address[/mask] [!] [port[:port]], -source [!] address[/mask] [!] [port[:port]]: 출발지주소를 지정한다. 포구이름이나 포구번호도 지정가능하다.

[!] [port[:port]]: 출발지의 포구이름이나 포구번호만을 지정할수 있다.

-d [!] address[/mask] [!] [port][:port]], -destination [!] address[/mask] [!] [port[:port]]: 목적지주소를 지정한다. 마찬가지로 포구이름이나 포구번호를 지정한다.

[!] [port[:port]]: 목적지의 포구이름이나 포구번호만을 지정할수 있다.

[!] typename : tcp/udp/icmp의 형태를 지정할수 있다.

-i [!] name[+], -interface [!] name[+] : 망의 대면부를 지정할수 있다. name+는 name이 공통으로 들어가는 모든것을 말한다.

-j target[port], -jump target [port] : 지정하는 목표(target)나 포구(port)로 전환(redirect)시킨다.

-m [+-]mark : 정합(match)되는 파케트에 수자를 할당한다.

-n, -numeric : IP주소와 포구번호를 수자 그대로 보여준다.

-l, -log : 일치되는 파케트를 핵심부리력(kernel log)에 남긴다 (sylog).

-v, -verbose : 상세한 정보를 보여준다.

-x, -exact : exact값을 보여준다.

-line-numbers : 규칙의 번호만을 보여준다.

-no-warnings : 경고통보문을 비활성화한다.

[!] -y, -syn : TCP파케트안의 SYN만을 설정한다.

-V —version : 패키지판본정보를 보여준다.

ipchains는 Linux Kernel 2.2.x판본에서 지원하는 파케트추출(packet filtering)형태의 방화벽기능을 관리하는 도구이다. 파케트려파기(packet filter)는 방화벽에서 파케트머리부(packet header)의 정보를 보기, 파케트의 통과(ACCEPT), 무시(DENY/REJECT) 등의 기능을 기본적으로 수행한다. 먼저 설정되여있는 chain방책(chain policy)이 있는가를 확인한다.

-L설정값은 chain방책을 보여준다.

```
kcc@linux:~
# ipchains -L
chain input (policy ACCEPT):
chain forward (policy ACCEPT):
chain output (policy ACCEPT):
```

Linux에서는 일반적으로 input, forward, output 등으로 chain방책이름을 기정부여한다. 우에서 보여주는것처럼 chain방책이 설정된것이 없다.

input chain방책에 출발지주소가 192.168.0.1인 규칙을 부여해보자.

```
# ipchains -A input -s 192.168.0.1
```

제대로 적용되였는가를 확인해보자.

```
kcc@linux:~
# ipchains -L
Chain input(policy ACCEPT):
target   prot   opt      source        destinailon    ports
-        all    ------   192.168.0.1   anywhere       n/a
Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
```

port는 procotol을 의미하며 source는 192.168.0.1, destination은 anywhere로서 0/0과 같다. ports는 설정되지 않았다.

만일 반대의 표현으로 192.168.0.1을 제외한다는 규칙을 부여할 때에는 -s ! 192.168.0.1에 !를 부여한다. 이것은 포구나 통신규약에도 사용가능하다.

이번에는 설정된 규칙을 -D chains규칙번호로 삭제해 보자. 여기에서 규칙번호(rule number)는 input chain방책의 입력순서로 -L input에 의해 출력되는 순서로 인식하면 된다.

```
# ipchains -D input 1
```

-L input 입력으로 input chain방책을 다시 보면 정확히 삭제된것을 확인할수 있다.

```
kcc@linux:~

# ipchains -L input
Chain input(policy ACCEPT):
```

이번에는 input chain방책에 출발지가 192.168.0.1인 파케트를 받는다는 규칙을 추가해보자.

```
# ipchains -A input -s 192.168.0.1 -j ACCEPT
```

다시 확인해보자.

```
kcc@linux:~

# ipchains -L input
Chain input (policy ACCEPT):
target        prot    opt        source
destination   ports
ACCEPT        all     ------     192.168.0.1
anywhere      n/a
```

우에서와 달리 target가 ACCEPT이면 파케트에 대한 허용을 의미한다. 만일 target가 DENY(거부)인 경우에 파케트가 없어진다. REJECT(거절) 역시 파케트가 없어지는것은 마찬가지이지만 그 반응을 출발지주소에 알리게 된다.

-s설정값은 출발지주소를 192.168.0.1로 지정하였다. 만일 목적지주소를 지정하면 -d address설정값으로 address에 주소를 입력한다.

-F설정값을 사용하면 chain방책의 모든 규칙을 삭제하게 된다.

```
kcc@linux:~

# ipchains -F
# ipchains -L
Chain input (policy ACCEPT):
Chain forward (policy ACCEPT):
Chain output (plicy ACCEPT):
```

아래의 간단한 레를 통해 보도록 하자.

```
kcc@linux:~
# ipchains  -A  input  -s  0/0  -d  0/0  139  -p tcp -y -j ACCEPT
# ipchains  -A  input  -s  0/0  -d  0/0  22  -p tcp -y -j ACCEPT
# ipchains  -A  input  -s  0/0  -d  0/0  80  -p tcp -t -j ACCEPT
# ipchains  -A  input  -s  0/0  -d  0/0  -I lo -j ACCEpt
# ipchains  -A  input  -s  0/0  -d  0/0  -I etho -j ACCEPT
# ipchains  -A  input  -s  61.40.233.124 53 -d 0/0 -p udp -j ACCEPT
# ipchains  -A  input  -s  168.126.63.1  53 -d 0/0 -p udp -j ACCEPT
# ipchains  -A  input  -s  0/0  -d  0/0  -p tcp -y -j DENY -l
# ipcheins  -A  input  -s  0/0  -d  0/0  -p udp -j REJECT-l
```

우선 input chain방책만을 사용하였으며 1, 2, 3번째에서 samba(139), ssh(22), 웨브봉사(80) 등은 syn파케트에 대해 모든 IP에서 접근가능하도록 허용(ACCEPT)하였다. 그리고 4, 5번째에서 lo와 eth0의 대면부접근을 허용하였다. 6, 7번째에서 령역이름봉사(name service)를 받기 위해 61.40.233.124.53, 168.126.63.1의 목적지주소를 허용하였다.

그중에서 령역이름봉사는 udp통신규약만을 사용하므로 -p설정값으로 udp 파케트만을 허용하도록 설정하였다.

8, 9번째에서 나머지의 출발지주소와 목적지주소로 오는 tcp통신규약을 가진 파케트는 모두 DENY하고 udp통신규약은 REJECT하는데 접근하는 파케트들은 이 설정값으로 체계리력(syslog)에 남기게 된다.

모든 ipchains방책을 입력한 경우, 이것은 주기억기에 적재되는것이므로 콤퓨터를 재기동하면 모두 사라지게 된다. 이것을 ipchains-save명령으로 보관할수 있다.

```
kcc@linux:~
# ipchains-save  > firewall_role
Saving input'.
Saving output'.
Saving forward'.
```

아래와 같이 ipchains-restore 명령으로 보관한 방책파일을 불러올수도 있다.

```
kcc@linux:~

# ipchains-restore < firewall_role
Restoring input'.
Restoring output'.
Restoring forward'.
```

【관련명령문】

ipchains-restore : 명령행에서 실행한 방책을 보관한다.

ipchains-save : 보관한 방책스크립트를 불러온다.

ipfwadm : kernel 2.0.x 판본에서 방화벽, 규칙설정, IP masquerading을 조정하는 핵심부도구이다.

Iptables : kernel 2.4.x 판본에서 방화벽, 규칙설정, IP masquerading 을 조정하는 핵심부도구이다.

참 고

masquerading(ipchains)

masquerading기법으로 한 콤퓨터에 두개의 망대면부를 설정한 다음 하나는 외부의 인터네트와 련결되여있고 다른 하나는 내부 게트웨이(192.168.0.101라고 가정)의 역할을 하도록 설정한다. 내부 게트웨이의 역할을 하는 대면부와 련결된 집선기(HUB)를 통해 같은 192.168.0.0/24대역의 IP를 설정하여 내부에서도 인터네트를 사용할수 있도록 할수 있다. 아래의 스크립트는 최소의 masquerading기법을 위한 입력으로 손쉽게 내부에 인터네트를 공유할수 있게 한다.

```
/sbi/ipchains -F
echo "1" > /proc/net/ipv4/ip_forward
/sbin/ipchains -A forward -s 192.168.0.1 -j MASQ
```

ipfwadm

/sbin/ipfwadm

Kernel 2.0.x판본의 방화벽관리도구

kernel 2.0.x판본의 IP방화벽, 통계분석관리도구이다.

형식 : ipfwadm -A 명령 매개변수[설정값]
ipfwadm -I 명령 매개변수[설정값]
ipfwadm -O 명령 매개변수[설정값]
ipfwadm -F 명령 매개변수[설정값]
ipfwadm -M[-l | -s][설정값]

범주 : 적용시키려고 하는 지령에 대하여 규칙범주를 선택하기 위해 다음과 같은 보조지령(flag)들이 사용된다.

-A [방향] : IP통계분석규칙으로서 선택적으로 방향을 명시할수 있다(in, out, both). 이 통계규칙은 출구 또는 입구파케트에 대해서만 수행하도록 지시한다. 기본 방향값은 both이다.

-I : IP입구방화벽규칙.

-O : IP출구방화벽규칙.

-F : IP전송방화벽규칙.

-M : P masquerading관리. 이 범주는 -l (보여주기) 또는 -s(timeout 시간설정)지령과만 조합하여 사용할수 있다.

설정값 : 다음의 설정값들은 수행할 특정행위를 명시한다. 특별한 언급이 없는 한 지령행에서 단 한가지만 사용가능하다.

-a [방책] : 선택한 목록에 대하여 하나 또는 그 이상의 규칙을 추가한다(Append).

-i [방책] : 선택한 목록의 맨 앞에 하나 또는 그 이상의 규칙을 삽입한다.

-d [방책] : 선택한 규칙목록으로부터 하나 또는 그 이상의 규칙을 삭제한다.

-l : 선택한 목록에 있는 모든 규칙을 보여준다.

-z : 선택한 목록의 모든 규칙에 대하여 파케트와 바이트계수기를 0 으로 돌려놓는다.

-f : 선택한 규칙목록을 지운다.
-p [방책] : 선택한 방화벽류형에 대한 기본방책을 바꾼다.
-s tcp tcpfin udp : masquerading에 사용되는 timeout값을 바꾼다.
-c : 선택한 방화벽류형에 대하여 IP파케트가 허가, 거부(deny) 또는 거절(reject)될것인가를 검사한다.
- n : 도움말을 보여준다.

매개변수 : 다음의 매개변수들은 추가, 삽입, 삭제, 검사 등의 지령들과 함께 사용할수 있다.
-P **통신규약** : 규칙 또는 검사할 파케트의 통신규약을 말한다.
-S **주소**[/mask] [port...] : 출발지주소와 포구를 지정한다.
-D **주소**[/mask] [port...] : 도착지주소와 포구를 지정한다.
-V **주소** : 파케트를 받거나 보내는 통로로 되는 대면부의 주소로서 선택적이다.
-w **이름** : 파케트를 주고 받는 대면부의 이름으로서 선택적이다.
-b : 쌍방향방식으로서 규칙을 쌍방향 모두의 IP파케트에 적용하도록 한다.
-e : 확장출력으로서 이 설정값을 주면 list지령에 대하여 대면부주소와 규칙설정값(존재하는 경우)을 추가로 보여준다.
-k : ACK비트가 설정된 TCP파케트에게만 해당된다.
-m : 전송허용하는 파케트에 대하여 masquerade한다.
-n : 수값출력. IP주소와 포구번호가 수값형태로 출력된다.
-o : 적합한 파케트에 대하여 핵심부체계접속(kernel login)을 하도록 한다.
-r [**포구**] : 파케트를 국부소케트로 전환(redirect)한다.
-t ANDmask XORmask : IP머리부의 TOS항목을 변경하기 위해 사용하는 마스크이다.
-v : 자세한 사항을 출력한다.
-x : 수자를 확장한다.
-y : SYN 비트가 설정되여있고 ACK 비트가 없는 TCP 파케트와만 부합된다.

현재 Kernel 2.0.x판본을 사용하고있는 Linux콤퓨터는 매우 적다고 보고 이 명령의 사용법에 대하여서는 생략하기로 한다. 그대신 ipchains나 iptables명령문의 사용법에 대하여 잘 알아둘 필요가 있다.

iptables

/sbin/iptables

Kernel 2.4.x판본의 방화벽관리도구

Kernel 2.4.x판본에서 방화벽, 규칙설정 IP masquerading을 조정하는 핵심부도구이다.

형식 : iptables -[ADC] chain 상세규칙 [설정값]
iptables -[RI] chain 규칙번호 상세규칙[설정값]
iptables -0 chain 규칙번호[설정값]
iptables -[LFZ] [chain][설정값]
iptables -[NX] chain
iptables -E 이전의 chain이름 새로운 chain이름
iptables -p chain target[설정값]
iptables -h

-A chain, -append chain : chain을 추가한다.

-D cahin, -delete chain : chain에서 규칙을 삭제한다.

-D chain 규칙번호, -delete chain규칙번호 : chain 방책내의 지정한 규칙번호를 삭제한다. 만일 규칙번호가 1이라면 chain규칙의 첫번째 규칙을 삭제한다.

-I chain [규칙번호], -insert chain [규칙번호] : chain방책의 지정한 수자번째에 삽입하거나 마지막에 규칙을 삽입한다.

-R chain 규칙번호, -replace : chain방책내의 지정한 수자번째의 규칙을 교체한다.

-L[chain], -list[chain]: 모든 chain방책이나 지정한 chain방책을 본다.

-F[chain], -flush: 모든 chain방책 또는 지정한 chain방책을 삭제한다.

-Z[chain], -zero: 모든 chain방책 또는 지정한 chain방책을 령으로 만든다.

-C chain, -check chain : 설정한 chain방책을 검사한다.

-N chain, -new-chain : 새로운 방책을 만든다.

-X[chain], —delete-chain : 사용자가 만든 chain이나 모든 chain을 삭제한다.

-P chain target, —policy chain target : chain 방책을 지정한 chain 방책으로 바꾼다.

-E old-chain new-chain, —rename-chain old-chain new- chain : chain명을 바꾼다.

-p, —procotol [!] proto : 통신규약을 지정한다. !은 제외의 의미이다.

-s, —source [!] addrees[/mask] : 출발지주소를 지정한다. mask는 C급이면 255.255.255.0이나 24bit로 표현된다.

-d, —destination [!] address[/mask] : 목적지주소를 지정한다.

-I, —in-interface [!] input name[+] : 수신하는 망대면부이름을 지정한다. name+은 name으로 시작하는 모든 대면부이름이다.

-j, —jump target : 지정한 target로 redirect시킨다.

-m : 지정한 match로 확장가능하다.

-n : IP주소와 포구번호를 수자그대로 보여준다.

-o, —out-iterface [!] output name[+] : 발신하는 망대면부이름을 지정한다.

-v, —verbose : 상세한 정보를 보여준다.

—line-numbers : 규칙방책을 보여줄 때에 행번호도 나태낸다.

-x, -exact : 정확한 값으로 나타낸다.

-V, —version : 판본정보를 보여준다.

iptables는 netfilter려과규칙에 사용한다. 이 명령은 ipchains와 사용법이 거의 같으나 가장 큰 차이점은 확장성에 있다. 보통 핵심부확장은 핵심부모듈하부등록부(/lib/modules/핵심부판본/kernel/net)에 존재하는데 iptables는 요구에 의하여 적재된다. 그러므로 자식모듈을 직접 적재할 필요는 없다. iptables의 확장은 공유서고형태로 보통 /usr/local/lib/iptables에 위치한다. 배포판은 이것을 /lib/iptables나 /usr/lib/iptabes에 넣게 된다. ipchain이 iptable에서 변경된 사항들은 다음과 같다.

• 미리 만들어진 chain이름(input, output, forward)가 소문자로부터 대문자로 바뀌였다.

- -i 지시자는 들어오는 대면부만 의미하며 INPUT와 FORWARD chain에서만 동작한다. FORWARD나 OUTPUT chain은 -o로 사용한다.
- TCP와 UDP포구는 --source-port, --sport(--destination-port, --dport)와 사용하게 된다. -p tcp 또는 -p udp설정값과 함께 사용되여야 한다.
- TCP -y 지시자는 -syn으로 바뀌였으며 -p tcp다음에 와야 한다.
- DENY target는 DROP으로 바뀌였다.

먼저 lsmod 명령으로 ip_tables모듈이 있는가를 확인한다.

```
kcc@linux:~

# lsmod
Module      Size     Usedby    Tainted:      P
smbfs       35552    2         (autoclean)
8139too     17440    2
ipx         16404    0         (autoclean)
ipchains    37704    0
ext3        61600    2         (autoclean)
jbd         40452    2         (autoclean)   [ext3]
usb-uhci    21764    0         (unused)
usbcore     51744    1         [usb-uhci]
```

ipchain 모듈이 있으므로 이것을 제거하고, ip_tables모듈을 적재해보자.

```
kcc@linux:~

# rmmod ipchains
# modprobe ip_tables
# lsmod
Module      Size     Usedby    Tainted:      P
ip_tables   11200    0         (unused)
smbfs       35552    2         (autoclean)
autofs      10948    0         (autoclean)   (unused)
8139too     17440    2
ipx         16404    0         (autoclean)
ext3        61600    2         (autoclean)
jbd         40452    2         (autoclean)   [ext3]
usb-uhci    21764    0         (unused)
usbcore     51744    1         [usb-uhci]
```

```
# iptables -A INPUT -s 0/0 -d 0/0 -dport 139 -p tcp --syn -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 -dport22 - p tcp -syn -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 --dport 80 -p tcp -syn -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 -i lo -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 -i eth0 -j ACCEPT
# iptables -A INPUT -s 61.40.233.12453 -d 0/0 -p udp -j ACCEPT
# iptables -A INPUT -s 168.126.63.153 -d 0/0 -p udp -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 -p tcp -syn -j DROP
# iptables -A INPUT -s 0/0 -d 0/0 -p udp -j REJECT
```

우선 INPUT chain 방책만을 사용하였고 1, 2, 3번째에서 samba(139), ssh(22), 웨브봉사기(80) 등은 syn파케트에 대해 모든 IP에서 접근가능하도록 허용(ACCEPT)하였다. 그리고 4, 5번째에서 lo와 eth0의 대면부접근을 허용하였다. 6, 7번째에서 이름봉사기를 받기 위해 61.40.233.124, 168.126.63.1의 목적지주소를 허용하였다. 그중에서 이름봉사기는 udp통신규약만을 사용하므로 -p설정값으로 udp파케트만을 허용하도록 설정하였다.

8, 9번째에서 나머지의 출발지주소와 목적지주소로 오는 tcp통신규약을 가진 파케트는 모두 거부(DENY)하며 udp통신규약은 거절(REJECT)한다.

모든 iptables방책을 입력하면 이것은 주기억기에 적재될뿐이므로 체계를 재기동한 후에는 모두 없어지게 된다. 이것을 iptables-save명령으로 보관할 수 있다.

```
# iptables-save > iptables_test
```

아래와 같이 iptables-restore명령으로 보관한 방책파일을 불러올수도 있다.

```
# iptables-restore < iptables_test
```

【관련명령문】

iptables-restore : 명령행에서 수행한 iptables방책을 보관한다.

iptables-save : 보관한 iptables방책을 불러온다.

참 고

masquerading(iptables)

ipchains명령과 마찬가지로 내부의 인터네트련결공유를 할수 있다.

masquerading기법으로 한 콤퓨터에 두개의 망대면부를 설정한 다음 하나는 외부의 인터네트와 련결되여있고 다른 하나는 내부게트웨이(192.168.0.1이라고 가정)의 역할을 하도록 설정한다. 내부게트웨이역할을 하는 대면부와 련결된 집선기(HUB)를 통해 같은 192.168.0.0/24대역의 IP를 설정하여 내부에서도 인터네트를 사용하도록 할수 있다. 아래의 스크립트는 최소의 masquerading기법을 위한 입력으로 손쉽게 내부에 인터네트를 공유할수 있다.

```
/sbin/iptables -F
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -s 192.168.1.2/24 -j MASQERADE
```

Solaris

Solaris는 BSD계렬의 SunOS에 토대하여 만들어진것으로서 System V계렬의 영향을 받은 SunOS 5판은 Solaris 2.x 라는 이름으로도 불리우게 되였다. 그리고 이후의 Solaris 2.7은 Solaris 7이라는 이름으로도 불리웠다. Sun회사는 이 OS를 자기 회사의 생산제품인 Sparc처리기상에서만 동작하는것으로 개발하였으나 Intel가동환경(Platform)이 널리 사용됨에 따라 Intel판도 함께 개발하였다. 그러나 본질적으로는 거의나 같은 원천프로그람에 토대하여 번역한 제품이므로 성능면에서의 큰 차이는 없다.

ispell

/usr/bin/ispell

철자법검사

주어진 본문파일에서 철자가 잘못 입력된 부분을 표시한다.

형식 : ispell[-설정값][파일]

-b : 파일을 여벌복사한다(back).
-d 사전파일 : 체계표준사전외의 지정한 사전을 리용한다.
-n : nroff/troff 파일을 검사한다.
-t : Tex, LaTex 파일을 검사한다.
-x : 여벌복사하지 않는다.
-h, -help : 도움말을 보여준다.
-v : 판본을 보여준다.

파일의 철자를 검사한다. 파일을 철자검사하여 체계의 사전에 수록된 내용과 맞지 않는 경우에 수정할 단어를 목록으로 만들어 선택할수 있게 해준다.

[주의]
기본적으로 영어에 한해서만 철자를 확인할수 있다.

ispell 명령문

명령문	내 용	명령문	내 용
i) ignore	변경안함	I) ignore all	모두 변경안함
r) Replace	변경	R) Replace all	모두변경
a) Add	사전에 단어추가	x) Exit	Ispell완료

아래에 《Have you had your brekfast?》라는 문장으로 이루어진 ispelltest 파일의 검사를 실례로 보여준다.

```
kcc@linux:~

# ispell ispelltest
Have you had your brekfast
1) breakfast      4) breakfast's
2) Breakfasts     5) breakfasted
3) breakfaster
```

우에서 보는바와 같이 철자가 틀린 단어는 다른 색바탕의 강조체(Highlight)로 표시되고 그 아래에 수정할수 있는 단어목록을 보여주어 선택할수 있게 하여준다. 알맞는 단어의 번호를 선택하면 해당한 단어가 수정되여 철자가 옳게 된다.

참 고

TeX

TeX는 일종의 조판프로그람이며 WYSIWYG방식을 지원하지 않는 문서편집기라고 할수 있다. 마치도 html을 사용하는것처럼 하나하나 꼬리표(Tag)를 할당하여 사용한다는 점에서는 불편감이 있지만 긴 문장을 편리하고 론리적으로 작성할수 있고 다양한 문자나 수식, 그림을 요구하는대로 삽입해넣을수 있다. 또한 전체의 양식을 쉽게 결정할수 있고 자주 사용하는 꼬리표를 간단히 줄인 형태로 다시 정의하여 사용할수 있는 등 다양한 기능들을 지원하고있다.

TeX는 사용하기가 쉽지 않으므로 일반적으로는 보다 편리한 LaTeX를 사용하고있으며 수식편집기능이 보다 강한 AMSTeX도 사용되고있다.

TeX는 문서형식을 일관하게 할수 있고 수학, 물리학, 공학 등에서 사용하는 수식을 정확명료하게 나타낼수 있으며 번역하지 않은 원천상태에서는 매우 용량이 작은것으로 하여 론문작성에 많이 사용되고있다. 또한 많은 Linux문서들이 TeX에 의해 만들어지고있다.

J

join /usr/bin/join

항목단위로 파일합치기

두 파일을 의미있는 형태로 묶어 하나의 파일로 만든다.

형식 : join [설정값]...파일 1 파일 2

-a 파일번호 : 지정한 파일을 기준에 맞게 합친다.
-e EMPTY : 빈 항목을 지정하여 공백을 나타낸다.
-i, —ignore-case : 비교대상항목이 다른 경우에 무시한다.
-j FIELD : 1FILED-2FILED 와 같다.
-j1 FILED : -1 FILED와 같다.
-j2 FILED : -2 FILED와 같다.
-o FORMAT : 초기화형식에 따라 합쳐진 결과를 보여준다. 1.2이면 《첫번째 파일의 2번째 항목》이라는 의미로 된다.
-t CHAR : 지정한 CHAR를 식별자로 리용한다.
-v 파일번호 : 지정한 파일을 기준으로 비교하여 다른 항목내용을 보여준다.
-1 FILED : 첫번째 파일의 지정한 항목번호를 기준으로 결합하여 보여준다.
-2 FILED : 두번째 파일의 지정한 항목번호를 기준으로 결합하여 보여준다.
--help : 도움말을 보여준다.
—version : 판본정보를 보여준다.

join은 두 파일을 의미있는 형태로 묶어 하나의 파일로 만드는 명령문으로서 공통으로 표시된 항목(FIELD)이 들어있는 행에 대응하여 합치게 된다.
실례로 아래와 같이 jointest파일과 jointest2파일이 있다고 하자.

```
kcc@linux:~

# cat jointest
1       one
2       two
3       three
4       four
5       five
# cat jointest2
1       1000
2       2000
4
5       5000
```

첫번째와 두번째 파일의 1번째 항목을 기준으로 공통된 내용을 join하여 보여준다.

```
kcc@linux:~

# join jointest jointest2
1 one 1000
2 two 2000
4 four
5 five 5000
```

항목기준을 -j1설정값 또는 -j2설정값으로 변경할수 있다. -j2 1 설정값은 두번째 파일의 1번째 항목을 기준으로 결합하는 설정값이다.

```
kcc@linux:~

# join -j2 1 jointest jointest2
1 one 1000
2 two 2000
4 four
5 five 5000
```

-v설정값은 주어진 2개의 파일을 비교하여 지정된 항목에 공통되지 않는 내용을 보여준다.

```
kcc@linux:~

# join -v 1 jointest jointest2
3 three
```

K

kbd_mode /usr/bin/kbd_mode

건반설정

현재의 건반방식을 확인하고 설정한다.

형식 : kbd_mode [설정값]

-s —scancode : 주사코드(scancode)방식으로 지정(RAW)
-k —keycode : 건코드(keycode)방식으로 지정(MEDIUMRAW)
-a—8bit : ASCII방식으로 지정(XLATE)
-u—unicode : UTF-8방식으로 지정(UNICODE)
-mode={8bit, keycode,Unicode,scancode} : 방식을 지정한다.
-h —help : 도움말을 보여준다.
-v —version : 판본을 보여준다.

이 명령문은 현재의 건반방식을 확인하고 설정할수 있다.

```
kcc@linux:~
# kbd_mode
The keyboard is in the default (ASCII) mode
```

ASCII방식으로 설정되여있다는것을 알수 있다. 설정된 내용을 변경하기 위해서는 우에서 설명한 설정값들을 참고로 하여 입력하면 된다.

kbdrate

/sbin/kbdrate

건반입력속도의 설정

건반의 건입력속도를 조절하여 건입력속도가 빠른 사람의 건입력오유를 줄이고 체감속도를 높여준다.

형식 : kbdrate[설정값]

-v : 판본을 보여준다.

-s : kbdrate를 아무것도 출력하지 않고 실행한다.

-r rate : 건반의 rate값을 변경한다. rate는 1초동안에 입력할수 있는 문자수를 말한다(rate의 범위 : 2.0~30.0).

-d delay : 건반의 delay 값을 설정한다. delay 시간은 하나의 건을 누르고있을 때 련속입력을 위한것인가를 구분한다. delay 시간은 1/1000 초단위를 사용한다(delay 범위 : 250~1000ms).

이 명령은 건반입력속도를 설정한다. 콤퓨터를 자주 사용하고 건입력속도가 빠른 사용자인 경우에 건반속도에 민감하게 반응하는것이 일반적이다.

일반사용자의 경우에는 차이를 느끼지 못하는 경우가 많다.

```
kcc@linux:~
# kbdrate
Typematic Rate set to 10.9 cps (delay= 250 ms)
```

건반이 1초에 입력할수 있는 문자수를 cps(Character Per Second)단위로 표시한다. 즉 우의 례에서는 1초당 10.9개의 문자를 입력할수 있다. delay는 사용자가 하나의 건을 누르고있을 때 그 입력이 련속된 문자의 입력을 위한것인가를 판단하는 시간을 말한다. 여기서는 250ms가 지나면 련속적인 입력을 위한것으로 간주하고 현재 누르고있는 건을 련속 입력하게 된다. 《rate 10.9 cps》와 《delay 250ms》는 IBM계렬콤퓨터에서 기본적으로 사용되는 표준이다.

kill /bin/kill

프로세스완료

프로세스를 완료하는 명령문이다.

형식 : kill[-s Signal][-a] pid … kill-l[Signal]

pid... : 완료시킬 프로세스ID나 프로세스이름을 지정한다.
-s : 특별히 보낼 Signal을 지정한다(Signal이름이나 번호로 지정하여준다).
-l : Signal로 사용할수 있는 Signal이름들을 보여준다. 이것은 /usr/include/linux/signal.h파일에서도 알수 있다.
-1, : -HUP프로세스를 재활성화한다.
-9 : 프로세스를 강제로 완료시킨다.

kill명령은 지정한 프로세스에 지정한 사건전달신호(Signal)를 보낸다. 이 명령은 체계에 문제가 발생하여 해당 프로세스를 말단에서 완료시키려고 하는 경우에 사용한다. 만일 kill명령으로 완료되지 않는 프로세스가 있는 경우에는 -9설정값으로 프로세스를 강제완료시킬수 있다.

먼저 ps명령으로 sshd프로세스를 확인한다. 뿌리사용자로서 PID 1765와 2287을 확인할수 있다.

```
kcc@linux:~
# ps auxlgrep sshd
root   1765  0.0  0.7  3520   912  ?      S  10:01  0:00  sshd
root   2287  0.0  0.3  2644   500  ?      S  11:46  0:00  sshd
root   2811  0.0  0.5  2496   688  pts/0  S  14:21  0:00  grep
sshd
```

ps aux명령을 입력하였을 때에 나타나는 프로세스정보의 매 항목의 내용은 다음과 같다(보다 상세한 내용에 대해서는 ps명령문부분을 참고할것).

root	USER	프로세스의 사용자
1765	PID	프로세스 ID
0.0	%CPU	마지막 1분동안 프로세스가 사용한 CPU점유률
0.7	%MEM	마지막 1분동안 프로세스가 사용한 주기억기의 점유률
3520	VSZ	가상기억기에 있는 프로세스의 KB단위의 크기
912	RSS	프로세스의 실지 주기억기의 크기로서 KB단위
?	TTY	련결되여있는 말단
S	STAT	실행되고있는 프로세스상태
10:01	START	프로세스가 시작된 날자
0:00	TIME	프로세스가 소비한 총시간
sshd	COMMAND	사용자가 실행한 명령이름

ps명령으로 확인된 프로세스정보는 kill PID명령으로 해당 프로세스를 완료시킬수 있다.

```
# kill 1765
```

만일 kill명령으로 완료되지 않는 경우에는 -9설정값으로 강제완료시킨다.

```
# kill -9 1765
```

kill -HUP pid명령으로 프로세스를 완료하였다가 다시 되살릴수 있다.

```
# kill -HUP 1765
```

【관련명령문】

Killall : 프로세스이름으로 프로세스를 완료하는 명령문이다.

참 고

Signal

Signal은 UNIX체계에서 프로쎄스사이의 통신을 하는 가장 오래된 방법으로서 프로쎄스에 비동기적인 사건(event)을 전달하는데 사용된다. 이와 같은 Signal은 건반새치기나 오유가 일어났을 때에 발생한다. 또한 쉘이 자식프로쎄스에게 작업명령을 보낼 때에도 사용된다. 이러한 Signal의 목록을 보기 위해서는 kill -l명령을 사용한다.

1) SIGHUP	2) SIGINT	3) SIGQUIT	4) SIGILL
5) SIGTRAP	6) SIGIOT	7) SIGBUS	8) SIGFPE
9) SIGKILL	10) SIGUSR1	11) SIGSEGV	12) SIGFEP
13) SIGPIPE	14) SIGALRM	15) SIGTERM	17) SIGCHLD
18) SIGCONT	19) SIGSTOP	20) SIGTSTP	21) SIGTTIN
22) SIGTTOU	23) SIGURG	24) SIGXCPU	25) SIGXFSZ
26) SIGVTALRM	27) SIGPROF	28) SIGWINCH	29) SIGIO
30) SIGPWR			

Signal의 개수는 체계마다 다를수 있다. 프로쎄스는 대부분의 Signal을 무시할수 있지만 19) SIGSTOP와 9) SIGKILL은 무시할수 없다. 앞에서 kill을 설명할 때에 뒤에 첨부되던 수자가 바로 이 Signal번호로서 대체로는 9번이나 19번을 사용한다.

프로쎄스가 Signal을 받으면 이 Signal을 블로크화할수도 있으며 직접 처리하거나 핵심부에게 넘길수도 있다. 핵심부에게 넘어가면 이 Signal은 기본동작대로 가능하게 된다.

Linux에서는 프로쎄스의 task_sturct에 보관된 정보로 Signal을 구현한다.

지원할수 있는 Signal의 개수는 처리기의 word크기에 제한을 받는다. 32bit word를 사용하는 체계에서는 32개까지의 Signal을 사용할수 있다. 처리대기중인 Signal은 signal항목에, 블로크화한 항목은 blocked항목에 들어간다. 블로크화된 Signal은 이 블로크를 해제할 때까지 대기상태로 남는다. 실제로 이 Signal이 움직이는 방식은 매우 자세한 설명을 필요로 하므로 Linux핵심부관련서적을 참고하기 바란다.

killall

/usr/bin/killall

프로세스완료

프로세스번호가 아닌 이름으로 프로세스를 완료하는 명령문이다.

형식 : killall[[-egiqvw][- Signal]이름...
killall-l
killall-V

-e : 매우 긴 이름에 대해 정확한 대응이름을 요구한다. 만약 지령이름이 15문자를 넘으면 완전한 이름은 리용불가능하다.
-g : 집단을 지정하여 프로세스를 완료시킨다. 같은 프로세스집단에 속한 여러개의 프로세스들이 발견되는 경우에도 siganl은 집단당 한번만 보내여진다.
-i : 프로세스완료전에 확인통보문을 보인다.
-l : 알려진 모든 Signal이름목록을 보여준다.
-q : 어떤 통보문도 보여주지 않는다.
-v : Signal을 제대로 보낸 경우, 상세한 정보를 보여준다.
-V : 판본정보를 출력한다.
-w : 프로세스가 완료될 때까지 대기한다.

killall명령문은 kill명령문과 달리 프로세스이름을 사용하여 프로세스를 완료할수 있다.

아래의 httpd 데몬을 확인해 보자.

```
kcc@linux:~

# ps auxlgrep httpd
root      857   0.0  1.1   81192   1452   ?  D    May27    0:00
/usr/sbin/httpd -
apache    887   0.0  2.3   78704   2988   ?  D    May27    0:00
/usr/sbin/httpd -
apache    888   0.0  2.3   78704   2988   ?  D    May27    0:00
```

```
/usr/sbin/httpd -
apache  889   0.0  2.3   78704   2988   ?   D    May27    0:00
/usr/sbin/httpd -
apache  890   0.0  2.3   78704   3000   ?   D    May27    0:00
/usr/sbin/httpd -
apache  891   0.0  2.3   78704   2988   ?   D    May27    0:00
/usr/sbin/httpd -
apache  892   0.0  2.3   78704   3000   ?   D    May27    0:00
/usr/sbin/httpd -
apache  893   0.0  2.3   78704   2992   ?   D    May27    0:00
/usr/sbin/httpd -
apache  896   0.0  2.3   78704   2992   ?   D    May27    0:00
/usr/sbin/httpd -
root    3440  0.0  0.5   2500    696   pts/0    S        21:53
0:00    grep  httpd
```

[주의]

같은 이름으로 여러개의 프로세스가 동작상태에 있는 경우, 프로세스번호가 낮은것을 완료시키면 나머지 프로세스도 함께 완료할수 있다.

killall뒤에 프로세스이름인 httpd를 입력하여 httpd데몬을 완료할수 있다.

```
kcc@linux:~

# killall httpd
# ps aux lgrp gttpd
root   3443  0.0  0.5  2496  688  pts/0 S 21:54 0:00 grep  httpd
```

httpd 데몬을 다시 실행해보자.

```
kcc@linux:~

# /etc/rc.d/init.d/httpd start
httpd를 시작합니다:                          [ 확인 ]
```

httpd 데몬을 확인한 다음에 소유가 apache소유임을 알수 있다.

```
kcc@linux:~

# ps aux lgrep httpd
root     3471   12.7 6.1  82344 7792 ?  S    21:55  0:00  /usr/sbin/httpd-
apache   3474   0.1  6.1  82344 7800 ?  S    21:55  0:00  /usr/sbin/httpd-
apache   3475   0.0  6.1  82344 7800 ?  S    21:55  0:00  /usr/sbin/httpd-
apache   3476   0.0  6.1  82344 7800 ?  S    21:55  0:00  /usr/sbin/httpd-
apache   3477   0.0  6.1  82344 7800 ?  S    21:55  0:00  /usr/sbin/httpd-
apache   3478   0.0  6.1  82344 7800 ?  S    21:55  0:00  /usr/sbin/httpd-
apache   3479   0.0  6.1  82344 7800 ?  S    21:55  0:00  /usr/sbin/httpd-
apache   3480   0.0  6.1  82344 7800 ?  S    21:55  0:00  /usr/sbin/httpd-
apache   3481   0.0  6.1  82344 7800 ?  S    21:55  0:00  /usr/sbin/httpd-
root     3483   0.0  0.5  2496  688  pts/0  S    21:55  0:00  grep httpd
```

killall -e설정값으로 프로세스관련집단을 지정하여 프로세스를 완료할수도 있다.

```
kcc@linux:~

# killall -e apache
# ps aux lgrep httpd
root   3486  0.0  0.5  2496  688  pts/0  S  21:55  0:00  grep
httpd
```

【관련명령문】

killpg : 프로세스집단에 Signal을 보내는 함수

kill : 프로세스를 완료하는 명령문

killall5 : 자신의 session을 제외한 모든 프로세스의 Signal을 보내어 실행중인 모든 프로세스를 완료한다.

ksyms

/sbin/ksyms

핵심부 symbol의 보기

핵심부symbol과 모듈에 대한 정보를 선별적으로 정리하여 보는 명령이다.

형식 : ksyms[설정값]

-a : 모든 symbol을 보여준다.
-h : column header를 보여주지 않는다.
-m : 모듈정보를 보여준다.

ksyms는 핵심부symbol에 대한 정보를 보여준다.

핵심부symbol은 /proc/ksyms파일을 보면 알수 있는데 ksyms명령문을 리용하여 원하는 내용을 선별적으로 볼수 있다.

핵심부symbol중에서 모듈에 대한 정보를 보기로 하자.

```
kcc@linux:~

# ksyms -m
Address     Symbol                              Defined by
C8860000    (18k)                               [8139too]
C8860000    _insmod_8139too_O/lib/modules/2.4.13-1hl/kernel/drivers/
net/8139too.o_M3BE47211_V132109                 [8139too]
c8864120    _insmod_8139too_S.data_L768         [8139too]
c8864120    _insmod_8139too_S.data_L768         [8139too]
c8863140    _insmod_8139too_S.rodata_L224       [8139too]
c8860060    _insmod_8139too_S.text_L12448       [8139too]

---------------이하 생략--------------------------
```

L

last /usr/bin/last

체계접속시간의 보기

사용자의 체계접속기록과 재기동기록을 보여주는 명령문이다.

형식 : last[설정값][사용자이름][tty..]

-num : 지정한 num만큼의 행만을 보여준다.
-n num : -num설정값과 같다.
-f file : 지정한 파일에서 정보를 불러들인다.
-R : 보여주는 목록에서 hostname항목은 보여주지 않는다.
-a : 보여주는 목록에서 hostname항목을 마지막 항목에 보여준다.
-d : 다른 호스트에서 접속한것만 보여준다.
-x : shutdown이 일어난 상태나 runlevel이 바뀐 상태를 보여준다.

/var/log/wtmp 파일은 체계를 기동한때로부터 현재까지의 모든 사용자의 체계접속(log in)과 체계탈퇴(log out)에 대한 정보를 가지고있다. 이것은 어떤 체계에 어떤 사용자가 망을 통해 접근하였고 탈퇴하였는가를 알수있게 하므로 체계관리에 매우 유용한 정보로 된다. 이 파일의 내용을 보여주는 명령문이 last명령문이다.《last 수자》설정값을 사용하면 last목록을 뒤로부터 지정한 수자만큼 보여준다.

```
kcc@linux:~
# last -9
root      pts/0    211.49.155.31    Tue May 28 21:53 still logged in
root      pts/0    211.255.253.57   Tue May 28 21:51-21:52 (00:00)
root      pts/0                Tue May 28 16:07 - 16:39 (00:32)
root      pts/3                Tue May 28 14:59    still logged in
root      pts/3                Tue May 28 14:59 - 14:59 (00:00
```

```
root      pts/1              Tue May 28 14:47 - 16:26 (01:39)
root      pts/1              Tue May 28 14:47 - 14:47 (00:00)
root      pts/4              Tue May 28 13:49 - 13:49 (00:00)
root      pts/4              Tue May 28 13:49 - 13:49 (00:00)
wtmp begins Fri Dec 14 12:57:45 2003
```

-R설정값을 입력하면 호스트이름을 제외한 나머지의 정보를 보여준다.

```
kcc@linux:~

# last -10 -R
root        pts/0       Tue May 28 21:53 still logged in
root        pts/0       Tue May 28 21:51   21:52 (00:00)
root        pts/0       Tue May 28 16:07   16:39 (00:32)
root        pts/0       Tue May 28 16:07   16:07 (00:00)
root        pts/3       Tue May 28 14:59 still logged in
root        pts/3       Tue May 28 14:59   14:59 (00:00)
root        pts/1       Tue May 28 14:47   16:26 (01:39)
root        pts/1       Tue May 28 14:47   14:47 (00:00)
root        pts/4       Tue May 28 13:49   13:49 (00:00)
root        pts/4       Tue May 28 13:49   13:49 (00:00)
```

매 리력파일(log file)들은 어느정도 용량이 차거나 시간이 지나면 새로운 리력파일을 생성하므로 이전의 리력기록들을 보기 위해서는 파일을 직접 지정할 필요가 있다. -f설정값은 리력파일을 열어 확인하는데 사용된다.

RedHat인 경우 리력파일은 /var/log 파일등록부에 생성된다.

wtmp.1파일은 이전의 리력파일이다.

```
kcc@linux:~

# last -f /var/log/wtmp.1 -10
japan     pts/5      211.49.153.31   Wed May 1 02:29    02:45 (00:16)
webmaste  pts/5      211.49.153.31   Wed May 1 02:29   02:29 (00:00)
root      pts/2      211.255.253.59  Tue Apr 30 22:34 gone    no logout
webmaste  ftpd18595  211.49.155.31   Tue Apr 30 22:22   22:34 (00:12)
webmaste  ftpd14099  211.49.155.31   Tue Apr 30 21:05   21:04 (00:06)
webmaste  ftpd13993  211.49.155.31   Tue Apr 30 21L04    21:04 (00:00)
```

```
webmaste  ftpd13944  211.49.155.31   Tue Apr 30 21:03 - 21:03 (00:00)
webmaste  ftpd13917  211.49.155.31   Tue Apr 30 21:03 - 21:03 (00:00)
webmaste  ftpd13898  211.49.155.31   Tue Apr 30 21:02 - 21:02 (00:00)
webmaste  ftpd13888  211.49.155.31   Tue Apr 30 21:01 - 21:02 (00:00)
wtmp.1 begins Mon Apr 1 09:03:12 2003
```

체계를 재기동한 내용의 정보만 보려는 경우에는 grep명령을 리용하여 검색한 결과를 출력한다.

```
kcc@linux:~

# last l grep reboot
reboot   system boot  2.4.13-1hl   Fri Jul 5 22:39    (30+23:08)
reboot   system boot  2.4.13-1hl   Fri Jul 5 22:16    (00:21)
reboot   system boot  2.4.13-1h1   Fri Jul 5 21:53    (00:00)
reboot   system boot  2.4.13-1h1   Fri Jul 5 21:53    (00:00)
```

우의 출력을 통해 재기동한 정보를 알수 있다. 첫행을 보면 체계의 핵심부판본과 재기동한 시간을 보여준다. 이 행의 끝부분에 있는 괄호안의 30+23:08은 체계가 재기동하기전까지 사용된 시간을 보여준다. 즉 이 체계는 재기동하기전에 30일 23시간 8분동안 동작하였다.

lastlog

/usr/bin/lastlog

접속리력의 보기

lastlog파일을 보여주는 명령문이다.

형식 : lastlog[설정값]

u 망접속명 : 지정한 망접속명의 lastlog정보만을 보여준다.
-t days : 지정한 날자기간에 망접속한 정보만을 보여준다.

lastlog명령은 /var/log/lastlog리력파일의 정보를 보여준다. 이 정보를 통해 사용자들이 마지막으로 체계접속(login)한 날자와 호스트명, 포구 등을 알수 있다. 만일 고급사용자나 일반사용자를 제외한 /etc/passwd에 등록된 사용자 혹은 이상한 식별이름의 망접속정보가 있는 경우에는 침입당한 흔적으로 볼수 있다.

lastlog명령으로 본 결과 root, admin, oracle 사용자만이 망접속한 흔적이 있다. 다른 사용자들은 한번도 체계접속한적이 없다.

```
kcc@linux:~

# lastlog
Username   Port     From            Latest
root       pts/0    211.49.155.31   화 5월 28 21:53:09 + 0900 2003
bin        **Never logged in**
daemon     **Never logged in**
adm        **Never logged in**
lp         **Never logged in**
sync       **Never logged in**
─────────────중간 생략────────────────────────────────
named      **Never logged in**
pcap       **Never logged in**
amanda     **Never logged in**
junkbust   **Never logged in**
mailman    **Never logged in**
ldap       **Never logged in**
```

```
pvm          **Never logged in**
admin        :0      화 5월 14 13:41:50 +0900 2003
oracle       :0      화 4월  2 08:12:58 +0900 2003
test          **Never logged in**
```

-u설정값을 리용하면 보려고 하는 사용자정보만을 출력할수 있다. 아래의 실례에서는 뿌리사용자가 접속한 포구와 자기의 IP, 마지막으로 접속한 시간을 보여준다.

```
kcc@linux:~

# lastlog -u root
Username Port  From     Latest
Root          pts/0  211.49.155.31 화 5월 28 21:53:09 +0900 2003
```

-t설정값을 입력하면 현재 시간부터 지정한 날자까지 최근에 체계접속한 사용자만을 뽑아볼수 있다. 아래에서는 마지막 체계접속날자가 현재로부터 14일이내인 사용자만을 보여준다.

```
kcc@linux:~

# lastlog -t 14
Username    Port    From           Latest
root        pts/0   211.49.153.31  화 5월 28 21:53:09 +0900 2003
```

-t 15를 입력하면 현재부터 마지막 체계접속날자가 15일이 지난 사용자인 root와 admin식별이름을 볼수있다.

```
kcc@linux:~

# lastlog -t 15
Username    Port    From           Latest
Root        pts/0   211.49.153.31  화 5월 28 21:53:09 +0900 2003
Admin       :0                     화 5월 14 13:41:50+0900 2003
```

【관련명령문】

last : /var/log/wtmp리력파일정보를 보여주는 명령문으로서 사용자의 체계접속, 체계탈퇴정보를 볼수 있게 한다.

who : /var/log/utmp 리력파일정보를 보여주는 명령문으로서 현재 체계접속한 사용자들을 보여준다.

참 고

POSIX (Portable Operating System Interface)

POSIX는 UNIX조작체계에 기초한 일련의 표준조작체계규약이다. 이것은 같은 규약에 준한 조작체계들의 호상간에는 프로그람을 새로 작성하지 않아도 어느 정도의 호환성을 보장하기 위해 만들어졌다.

POSIX내의 매 표준은 POSIX라는 용어다음에 소수로 표시하도록 되여있다. 실례로 POSIX.1은 C언어응용프로그람대면부의 표준이며 POSIX.2는 표준의 쉘과 편의프로그람에 관한것이다. 이외에도 주요한 규약들이 개발되고있다. 이와 같은 POSIX규약은 IEEE의 후원하에 개발되고있다.

BSD(Berkeley Software Distribution : Berkeley Software Design)

BSD는 UC버클리에서 개발배포하는 UNIX를 말한다. BSDUNIX는 많이 사용되여왔으며 적지 않은 UNIX체계들이 이에 토대하여 만들어졌다. 이 계렬의 대표적인것으로는 FreeBSD, NetBSD, OpenBSD 등을 들수 있다.

ldconfig

/sbin/ldconfig

동적련결의 설정

공유서고 cache를 재설정하는 명령문이다.

형식 : ldconfig[설정값]등록부

-C cache : /etc/ld.so.cache파일대신에 지정한 파일을 리용한다.
-f conf : /etc/ld.so.conf파일대신에 지정한 파일을 리용한다.
-l : 수동으로 개개의 서고를 련결(link)한다.
-n : 지령행에서 지정한 등록부만을 진행하며 /usr/lib, /lib,/etc/ld.so.conf파일은 진행하지 않는다.
-N : cache를 갱신하지 않고 련결만을 갱신한다.
-p -print-cache : 현재 cache에 보관된 등록부와 서고목록을 보여준다.
-r root : 지정한 root를 뿌리등록부로 리용한다.
-V, —verbose : 현재의 판본정보와 관련된 등록부와 련결들을 상세히 보여준다.
-V, —version : 판본정보를 보여준다.
-x : 련결을 갱신하지 않고 cache만을 갱신한다.
-V, —help, —usage : 사용법을 보여준다.
-V, —version : 판본정보를 보여준다.

서고란 체계에서 프로그람을 직접 번역할 때에 프로그람들에서 공통으로 사용하는 기능을 미리 모아놓은 파일이다. 이것은 프로그람들로 하여금 직접 하나하나의 기능을 만들어주는 대신에 이미 존재하는 서고의 기능을 활용할 수 있게 한다. 이것을 동적련결프로그람(dynamic linked program)이라고 부른다. 또한 번역할 때에 자체서고를 사용하는것을 정적련결프로그람(statically linked program)이라고 한다.

동적으로 련결된 실행파일은 공유서고에 매우 의존적이지만 현재의 대부분의 서고판본들에 한해서는 비의존적이라고 할수 있다(glibc-2.1판으로부터 glibc-2.2판으로 갱신된것과 같은 큰 변화가 있는 경우에는 서고판본에 의존적일수 있다).

새로운 판본의 서고를 설치할 때에는 그를 사용하기 위해 특별한 파일등록부에 설치하고 ldconfig를 실행하여 공유서고 cache를 다시 설정해야 한다.

ldconfig는 존재하는 파일을 조사하여 /etc/ld.so.cache의 cache파일을 설정하면서 필요한 서고를 적재하고 기호련결(Symbolic Link)을 만드는 역할을 하는것이다.

먼저 /etc/ld.so.conf파일을 보자.

```
kcc@linux:~
# cat /etc/ld.so.conf
/usr/kerberos/lib
/usr/X11R6/lib
/usr/lib/qt-2.3.2/lib
/usr/lib/sane
/usr/lib/mysql
/usr/lib/qt-1.45/lib
/usr/lib/wine
/usr/lib/gtk
```

ld.so.conf에서 빠진 내용은 home등록부의 .bash_profile이나 bashrc에 있는 LD_LIBRARY_PATH환경변수에 등록부를 추가한다.

아래의 실례는 oracle 식별이름에서 oracle설치를 위한 LD_LIBRARY_PATH를 추가한 내용을 보여준다.

```
kcc@linux:~
# cat /home/oracle/.bash_profile | grep LD_LIB
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/
lib:$ORACLE_HOME/jdbc/lib
```

공유서고의 내용이 변경된 경우에는 ldconfig 명령을 실행하여 공유서고 cache를 다시 만들도록 한다.

```
# ldconfig
```

ldd

/usr/bin/ldd

서고의존성을 보기

공유서고의 의존성을 확인하는 명령문이다.

형식 : ldd[설정값]파일

—version : 판본정보를 보여준다.

-v, —verbose : 관련된 상세한 정보를 보여준다.

-d, —data-relocs : 있어야 할 곳에 없는 오브젝트의 위치를 다시 배치하고 결과를 보여준다.

-r, —function-relocs : 자료오브젝트와 함수를 다시 배치하고 오브젝트나 함수가운데서 찾을수 없는 결과를 보여준다.

—help : 사용법을 보여준다.

ldd명령은 프로그람이 사용하는 공유서고의존성을 알수 있게 한다. 례를 들어 ls명령문의 공유서고목록을 보기로 하자.

```
kcc@linux:~
# ldd /bin/ls
        libtermcap.so.2 => /lib/libtermcap.so.2 (0x4002c000)
        libc.so.6 => /libc.so.6(0x40030000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

우의 내용에서 lib/libc.so.6은 C서고이다. 다른 프로그람작성언어들에서도(적어도 그 언어의 서고를 구현하기 위해) C서고를 사용하는 경향이 있다.

이로부터 많은 프로그람들에 이 서고가 포함되게 된다.

/lib/ld-linux.so.2는 다른 모든 서고를 적재한다.

less /usr/bin/less

파일보기

파일내용을 페지단위로 보여주는 명령문으로서 more명령의 발전된 형태이다.

형식 : less[설정값][파일이름]

-?, -help : 명령문에 대한 도움말을 보여준다.

-a, -search-skip-screen : 마지막행을 화면에 출력한 다음 탐색을 시작한다.

-c, -clear-screen : 필요할 때에 전체화면을 우에서 아래로 다시 갱신한다.

-C, -CLEAR-SCREEN : -c설정값과 기능이 같지만 갱신할 때에 화면전체를 지우고 시작한다.

-e, -quit-at-eof : 파일의 끝에 두번째로 도달하면 자동적으로 완료한다. 기본값으로 q가 입력되여야 완료된다.

-E, -QUIT-AT-EOF : 파일의 끝에 도달하기만 하면 자동적으로 완료한다.

-i ignor-case : 찾기패턴의 단어자체의 대소문자를 구분함이 없이 탐색한다.

-l, -IGNORE-CASE : 찾기패턴에 포함된 단어의 대소문자를 구분함이 없이 탐색한다.

-m, -long-prompt : 화면아래부분의 입력독촉(prompt)부분에 more지령과 같이 전체 파일내용에서 파일을 읽은 %수를 표시한다.

-M, -LONG-PROMPT : 화면아래부분의 입력독촉에 more지령보다 상세한 정보를 보여준다.

-n, -line-numbers : 행번호를 보여주지 않는다.

-N, -LINE-NUMBERS : 각 행의 시작하는 부분에 행번호를 보여준다.

-ofileame, —log-file-filename : 지정한 filename에 보여지는 내용을 보관한다.

-Ofilename, -LOG-FILE=filename : -o설정값의 기능과 비슷하지만 filename이 존재하면 확인절차가 없이 filename에 내용을 덧쓰기한다.

-q, -quit or -silent : 특정오유가 발생하여도 체계경고음을 내지 않는다.
-Q, -QUIET or -SILENT : 어떤 경우에도 체계경고음을 내지 않는다.
-s, -squeeze-blank-lines : 련속되는 공백행을 하나의 행으로 처리한다.
-xn, -tabs=n : 지정한 n값만큼 Tab간격을 조정한다. 기본값은 8이다.
-V, -version : 판본정보를 보여준다.

less명령은 more보다 기능이 추가된 페지단위로 파일의 내용을 보여주는 명령문이다.

/etc/sendmail.cf파일을 한번 보자. less명령을 실행한 후에 입력되는 대부분의 명령문은 vi의 기능과 매우 류사하다. 실행후에 h를 입력하여 사용가능한 명령을 보자.

```
kcc@linux:~
# less /etc/sendmail.cf
------------------------------------------------------------------
                 SUMMARY OF LESS COMMANDS

   Commands marked with * may be preceded by a number, N.
   Notes in parentheses indicate the behavior if N is given.

h H            Display this help.
q :q Q :Q ZZ    Exit.
------------------------------------------------------------------
                        MOVING

e ^E j ^N CR * Forward one line (or  N lines).
y ^Y k ^K ^P * Backward one line (or N lines).
f ^F ^V SPACE * Forward one window (or N lines).
b ^B ESc-v      * Backward one window (or N lines).
z             * Foward one window (and set window to N).
w             * Backward one window(and set window to N).
ESC-SPACE     * Forward one window,but don't stop at end-of-file.
d ^D          * Forward one half-window(and set half-window to N).
u ^U          * Backward one half-window (and set half-window to N).
HELP -- Press RETURN for more, or q when done
```

less명령문

페지이동	Space bar, ctrl+V, F, ctrl+F
z	Space bar와 비슷하지만 수자를 입력하고 z를 입력하면 지정한 수자행만큼 이동한다.
한행 아래로 이동	Enter건, Ctrl+N, e, Ctrl+E, j
반페지 아래로 이동	D, ^D
반페지 우로 이동	B, ^B, ESC-v
한행 우로 이동	Y, Ctrl+Y, Ctrl+P, k, Ctrl+K
반페지 우로 이동	u, Ctrl+U
화면을 새로 고침	r, Ctrl+R, Ctrl+l
완료	q

이외에도 파일찾기를 위해서는 /이나 ?를 입력한 다음에 찾으려고 하는 패턴을 찾으면 된다. /는 아래로 이동하면서 찾기를, ?는 우로 이동하면서 찾기를 한다. 하나의 패턴을 찾으면 찾은 문자에 대해 블로크가 지정되며 다음것을 찾기 위해서는 n건을 입력한다.

이외에도 많은 기능이 있지만 기타 기능은 vi편집기와 매우 류사하므로 vi편집기부분을 참고하도록 한다.

참 고

GRUB(GRand Unified Bootloader)

GRUB는 Linux와 BSD계렬, GNU/Hurd는 물론이고 다양한 OS들에서 사용할수 있는 기동관리자(boot manager)이다. Linux에서만 사용할수 있는 LILO와는 달리 알려진 많은 체계들에서 설치 및 설정을 할수 있다. 또한 엄밀한 의미에서의 기동관리자라고 할수 없는 Loadlin에 비해서도 완성된 기동관리자라고 할수 있다.

lilo

/sbin/lilo

Linux기동적재기

처음 기동을 시작할 때에 실제적으로 기동과정을 담당할 환경변수(BOOT_IMAGE)를 불러오기 위해 먼저 구동되는 명령이다.

형식 : lilo[설정값]

-v : 상세한 정보를 보여준다.

-q : 현재 map파일의 목록을 보여준다. lilo는 기본적으로 /boot/map에 의해 유지되고 boot에 대한 핵심부의 이름과 위치를 포함한다. 이 설정값은 그곳에 있는 파일들의 목록을 보여준다.

-m map-file : 지정한 map-file을 사용한다.

-C config-file : 지정한 confile-file에서 설정파일을 읽어온다. 기본은 /etc/lilo.cof 파일이다.

-d delay : lilo boot화면에서 아무 건입력이 없는 경우, 자동적으로기정체계로 기동한다. 자동적으로 기동되는 시간을 delay로 지정한다. 이것은 0.1초단위로 입력하며 입력가능한 기록권(label)은 Tab건을 입력하면 볼수 있다. 만일 delay가 50으로 설정되였다면 5초를 의미한다.

-D label : lilo boot화면에서 기정적으로 선택될 label을 지정한다.

-r root-directory : 어떤 작업을 하기전에 chroot로 등록부를 변경한다. 기동디스크로 기동하여 설정을 변경하는 경우에 사용한다.

-t : 새로운 기동분구(boot sector)나 map-file을 생성하지 않고 검사만을 진행한다.

-c : map를 간단하게 하여 린접한 분구로부터의 요청을 하나로 한다.

-f disk-tab : 지정한 disk-tab파일을 디스크 geometry매개변수로 한다. 기본값은 /etc/disktab이다.

-i boot-sector : 지정한 boot-sector파일로 새로운 기동분구를 생성한다. 기본값은 /boot/boot.b이다.

-l : sector/head/cylinder주소대신에 선형분구주소(linear sector address)를 생성한다.

-s save-file : lilo가 기동분구를 덧쓰기할 때에 기본값 /boot/boot.NNNN으로 이전의 내용을 보관한다. NNNN은 장치에 의존한다. 이 설정값은 그 대신 기동분구를 보관하기 위해 지정한다.

또는 -u와 같이 지정하여 기동분구를 복구하기 위해 지정한다.
-S save-file : 지정한 파일로 덧쓰기를 한다.
-u device-name : 기동분구에 보관된 lilo를 제거한다.
-U device-name : -u와 같이 lilo를 제거하는것은 동일하지만 시간-지표를 검사하지 않는다.
-R command line : 다음 기동시에 사용되는 기본지령을 설정한다.
-I label : 체계가 시작된 후에 환경변수 BOOT_IMAGE의 실행핵심부의 label을 참조할수 있도록 한다.
-V : 판본정보를 보여준다.

lilo는 GRUB와 함께 Linux의 기동적재기중의 하나이다.
기본설정파일은 /etc/lilo.conf파일이다.
아래에 설정파일의 실례를 보여주었다. 체계의 하드디스크에 설치한 OS내용은 다음과 같다. Windows체계는 Primary Slave HDD 첫번째 구획에 설치(/dev/hdbl)되여있으며 Linux는 Primary Master HDD 8번째 구획에 /root설치(dev/hda8)되여있다고 가정하자.

kcc@linux:~

```
# cat /etc/lilo.conf
prompt
timeout=50
default=linux
boot=/dev/hda
map=boot/map
install=boot/boot.b
message=/boot/message
lba32

image=boot/vmlinux-2.4.13-1hl
        label=linux
        read-only
        root=dev/hda8
other=/dev/hdb1
        label=windows
```

이 파일에서 설정한 설정값을 간단히 보고 넘어가자.

- timeout: 기동때에 LILO화면에서 건반입력이 없는 경우에 자동기동되는 시간을 지정한다.
- default: 다중기동을 설정하는 경우에 기정적으로 선택될 label을 지정한다. 우의 례에서 Linux는 /dev/hda8에 설치되여있으며 Windows는 /dev/hdb1구획에 설치되여있다. 기정적으로 linux를 지정하여 label설정값에서 지정한 이름으로 Linux를 지정하였다. 만일 이것을 Windows로 변경하면 기정선택은 Winows로 된다.
- boot: LILO가 설치될 구획을 지정한다. /dev/hda라고 지정하면 Primary Master HDD의 MBR령역에 LILO가 설치된다.
- map: map image를 지정한다.
- message: LILO screen image를 지정한다.
- lba32: HDD의 LBA기능을 사용한다. 대용량의 하드디스크에서 사용한다.
- image: Linux의 kernel image를 지정한다.
- label: 이름을 지정한다.
- read-only: 읽기전용으로 적재한다.
- root: LILO가 읽어들일 기본구획을 지정한다.
- other: 다중기동(multi-boot)으로 설정할 구획을 지정한다.

참 고

lilo가 시작되지 않을 때

LILO가 시작되지 않을 때 LILO가 MBR에 제대로 설치되지 않은 상태, Linux를 설치할 때 LILO설정값을 선택하지 않았거나 기동분구의 위치를 알아낼수 없는 경우

- L: 하드디스크의 결함이 있을 때에 발생하는 통보문.
- LI: Map installer실행이 없이 /boot/boot.b가 이동했을 때에 발생하는 문제로서 설치때에 BIOS에서의 장치형식과 설치후 BIOS에서의 장치형식이 다를 때에 발생한다.

• LIL: Map파일에서 descriptor table을 적재하지 못할 때 발생하며 하드디스크의 결함으로 인하여 발생하는 경우가 많다.

• LIL?: 두번째 기동분구적재기(boot sector loader)가 잘못된 주소에 적재되였을 때에 발생하는 오유.

• LIL-: descriptor table이 파괴되였을 때에 발생한다.

LILO나 GRUB 등의 기동적재기가 사라졌을 때에 기동CD로 복구하기

LinuxCD를 CD-ROM구동기에 넣으면 《boot :》라고 화면의 왼쪽아래에 출력된다. 이때 아래와 같이 입력한다.

```
boot: vmlinuz root-/dev/hda()
```

⚠ 경 고

여기서 ()는 (root partition)이 있는 구획이다. Boot로 구획을 나눈 경우에는 /boot구획번호를 지정하며 /partition만 있는 경우에는 /구획번호를 입력한다.

기동이 완료되면 뿌리식별이름으로 체계접속하고 LILO설정파일을 체계에 맞게 수정한다.

```
# vi /etc/lilo.conf
```

수정후에 lilo 명령을 입력하여 lilo를 갱신한다.

```
# lilo
```

⚠ 경 고

이 지령을 주지 않으면 수정한 내용이 체계에 적용되지 않는다. 핵심부콤파일을 하고 lilo.conf를 수정하여도 이 지령이 없는 경우에는 새로운 kernel image로 기동할수 없다.

표준 기동선택안내문을 바꾸기

표준 기동선택안내문(boot select menu)을 바꾸는데는 두가지의 방법이 있다.

첫번째 방법은 명령행에서 -D설정값으로 안내문선택을 바꾸는것이다. 이때 《windows》는 Windows체계의 label명이다.

```
# lilo -D windows
```

다른 하나의 방법은 /etc/lilo.conf파일에서 default설정값을 Windows의 label로 수정한 후에 LILO를 갱신하는것이다.

```
kcc@linux:~

───────────
default=windows
───────────
# lilo
Added linux
Added windows *
```

우의 통보문에서 *가 windows로 선택된것을 볼수 있다.

디스크의 MBR령역에서 기동적재기(boot loader)를 삭제하기

DOS상태에서 fdisk명령을 아래와 같이 실행한다.

```
# fdisk /mbr
```

ln

/bin/ln

파일련결명령문

파일을 실지의 경로가 아니라 사용하기 편리한 다른 경로로 접근할수 있도록 지정하는 명령문이다.

형식 : ln[설정값]원본[대상]
ln[설정값]원본...등록부

-b —backup : 대상파일이 있는 경우에 예비보관파일을 생성한다.

-d, -F, —directory : 파일등록부에 하드련결(Hard Link)이 생성되게 한다(root식별이름만 기능).

-f, —force : 련결(link)을 생성할 대상파일이 있는 경우에도 강제적으로 새로운 link를 생성한다.

-I, —interative : 련결을 생성할 대상파일이 있는 경우에 삭제유무를 사용자에게 문의한다.

-n, —no-dereference : 련결(link)할 원본이 기호파일이면 그 기호파일의 대상파일을 추적하여 련결한다.

-s, —symbolic : 련결할 원본이 기호파일이면 기호파일을 link한다.

-v, —verbose : 진행되는 상태를 자세하게 보여준다.

—help : 도움말을 보여준다.

—version: 판본정보를 보여준다.

-S, —suffix backup-suffix : link를 생성할 대상파일이 이미 있는 경우, 이전의 대상파일을 예비보관할 파일확장자를 지정한다.

-V, —version-control {nurnbered. existing, simple} : 예비보관하는 방법을 지정한다.

t, numbered : 항상 번호로 된 예비보관파일을 만든다.

nil existing : 대상파일이 있는 경우에만 예비보관파일을 만든다.

never, simple : 간단한 예비보관판을 만든다.

파일을 련결(link)하는 방법에는 기호련결과 하드련결이 있다. 기호련결을 쏘프트련결이라고도 한다.

기호련결, 쏘프트련결

일반적으로 련결(link)이라고 하면 거의나 기호련결(Symbolic link)을 의미한다. 기호련결은 쏘프트련결(Soft link)이라고도 한다. 기호련결은 불필요한 파일의 복사를 하지 않아도 되게 한다. 보통 여러 파일등록부에서 꼭 같은 서고를 요구하는 경우나 하나의 파일을 여러 사용자가 공통으로 사용하는 경우에도 많이 쓰인다.

```
kcc@linux:~
# ln -s /home/admin/html /var/www
# ls -la www
lrwxrwxrwx 1 root root 7 May 7 11:38 www->/home/admin/html
```

하드련결

하드련결(Hard link)은 똑같은 파일크기로 원본 파일이 수정되는 경우, 하드련결된 파일도 원본과 동일하게 변경되며 항상 같은 내용을 유지할수 있게 한다. 원본이 삭제되여도 원본과 같은 내용을 가지고있으므로 자원을 공유하면서 자료를 안전하게 관리하려고 할 때에 효과있게 사용할수 있다.

```
kcc@linux:~
# ls -al test
-rw------   1   root   root   932   May   7   11:40   test
# ln test test_hard link
# ls -al test*
-rw------   1   root   root   932   May   7   11:40   test
-rw------   1   root   root   932   May   7   11:40
test_hardlink
# rm test
rm : remove 'test'? y
# ls -al test*
-rw------   1   root   root   932   May   7   11:40
test_hardlink
```

기호련결과 하드련결의 차이

	기호련결	하드련결
파일크기	작다	원본과 같다
원본파일이 삭제되였을 때	사용불가능하다	사용가능하다
권한	'l'로 표시된다	일반파일이다

참　고

하드디스크가 완전히 찼을 때

/home등록부가 완전히 찼는데 새로 하드디스크를 대책할 시간이 없는 경우에는 ln을 리용하여 이러한 곤난상태를 일시적으로 타개할수 있게 하는 방법을 아래에 설명하였다. 먼저 현재 매 구획의 남은 용량을 본다.

```
kcc@linux:~

# df
Filesystem   1k-blocks    Used  Available  Use%   Mounted on
/dev/hda1     2063504  1301408     657276   67%   /
/dev/hda5      505605   500548       5057   99%   /home
/dev/hda3     2063536   712716    1245996   37%   /opt
none            63244        0      63244    0%   /dev/shm
/dev/hda2     2577424  1749912     696584   72%   /usr
```

식별이름이 있는 구획이 완전히 차면 체계접속(log in)할수 없는것은 당연하다. 그러나 우의 경우에 /opt가 있는 /dev/hda5구획은 용량이 충분하다는것을 알수 있다. 이때에는 사용자구획의 파일등록부중 몇개를 /opt아래에로 옮기고 기호련결을 걸어주기만 하면 림시로 용량을 확보할수 있다. 이러한 림시대책에도 한계가 있으므로 우와 같은 상태에서는 빨리 하드디스크를 새로 대책하거나 불필요한 자료를 삭제하여 용량을 확보하도록 하는것이 보다 효과적이다.

locale

/usr/bin/locale

지역정보보기

언어를 비롯한 매 나라별로 선택가능한 정보를 볼수 있다.

형식 : locale[설정값]이름
locale[설정값][-a|-m]

-a, —all-locales : 사용가능한 지역(locale)정보의 이름을 보여준다.
-m, —charmaps : 사용가능한 문자지도(charmaps)의 이름들을 보여준다.
-C, —category-name : 사용가능한 범주의 이름을 보여준다.
-k, —keyword-name : 선택된 실마리어(keyword)의 이름을 보여준다.
-v, —verbose : 상세한 정보를 보여준다.
-?, —help : 도움말을 보여준다.
-V, —version : 프로그람의 판본정보를 보여준다.

지역(locale)정보는 프로그람의 언어사항과 관련있다. 다양한 지역정보를 설치해둔 경우에는 다음의 환경변수설정을 통하여 지역을 지원하는 프로그람의 실행방식을 선택할수 있다. 기본지역정보는 C 또는 POSIX로 설정되여있다. 현재 체계의 지역정보는 locale명령으로 볼수 있다.

```
kcc@linux:~
# locale
LANG=ko_KP.eucKP
LC_CTYPE="k_KP.eucKP"
LC_ NUMERIC="ko_KP.eucKP"
LC_TIME="ko_KP.eucKP"
LC_COLLTAE="ko_KP.eucKP"
LC_ MONETARY="ko_KP.eucKP"
LC_MESSAGES="ko_KP.eucKP"
LC_PAPER="ko_KP.eucKP"
LC_NAME="ko_KP.eucKP"
LC_ADDRESS="ko_KP.eucKP"
LC_TELEPHONE="ko_KP.eucKP"
LC_MEASUREMENT="ko_KP.eucKP"
```

RedHat계렬의 배포판에서 X Window의 지역관련 파일등록부는 /usr/X11R6/lib/X11/locale이다.

```
kcc@linux:~

# ls /usr/X11R6/lib/X11/locale
C               iso8859-1     iso8859-7     koi8-u          tscii-0
armscii-8       iso8859-10    iso8859-8     looale.alias    vi_VN.tcvn
common          iso8859-13    iso8859-9     locale.dir      vi_VN.viscii
compose.dir     iso8859-14    iso8859-9e    microsoft-cp1251   zh
en_US.UTF-8     iso8859-15    ja            microsoft-cp1255   zh_CN
georgian-academy              iso8859-2     ja.JIS          microsoft-cp1256
zh_CN.gbk
georgian-ps     iso8859-3     ja.SJIS       mulelao1        zh_HK.big5
ibm-cp1133      iso8859-4     ko            nokhchi-1       zh_HK.big5hkscs
iscii-dev       iso8859-5     koi8-c        tatar_cyr       zh_TW
iscii-3342      iso8859-6     koi8-r        th_TH           zh+TW.big5
```

locate

/usr/bin/locate

파일위치찾기

자체의 자료기지를 통해 파일을 찾아주는 명령문이다.

형식 : locate[설정값]
slocate[설정값]

파일을 찾는 명령문인 locate는 현재 거의 사용되지 않으며 이 locate에 보안을 강화한 slocate가 많이 사용되고있다. 현재 사용되는 locate는 대부분 slocate에 기호련결로 만들어진 파일이다. 자세한 내용은 slocate명령문 설명에 언급되여있으므로 그를 참고할수 있다.

【관련명령문】

find : 파일을 찾을 경로와 찾으려고 하는 파일의 표현식에 의해 파일을 찾는 명령문이다.

slocate: locate 명령문의 보안강화판이다.

lockfile

/usr/bin/lockfile

신호기파일생성기

procmail에서 접근조종파일을 생성하는 명령문이다.

형식 : lockfile[설정값]파일이름

-sleeptime : lockfile이 생성시도에 실패한 다음 다시 시도하기까지의 시간을 sleeptime으로 한다. 기본값은 8초이다.

-r retries : 파일생성을 위한 재시도 한 다음 새로 시도할 회수를 지정한다. 기본값은 -1로써 정지하지 않고 계속 시도한다.

-l locktimeout : lockfile이 마지막으로 수정/생성된 다음 지정한 locktimeout시간이 지나 강제적으로 lockfile을 제거한다.

-s suspend_time : 지정한 suspend_time 다음 lockfile이 삭제된다. 기본값은 16초이다.

-! : 결과값의 역이다. 주로 쉘스크립트에서 사용한다.

-ml, -mu : mail spool 등록부권한이 허용되는 경우에는 체계의 mail box 를 lock(-ml), 또는 unlock(-mu) 시킬수 있다.

lockilfe 명령은 procmail패키쥐중의 하나로서 특정한 파일, 장치, 자원에 대해 접근통제를 하여주는 신호기파일(semaphore file)인 잠금파일을 생성하는데 사용한다. 이 잠금파일은 특정한 파일, 장치, 자원을 특정한 프로쎄스가 사용하고있다는 신호기발과 같은것으로 사용하여 다른 프로쎄스의 접근을 제한한다. 잠금파일은 기본적으로 /var/lock에 위치한다.

참 고

신호기(semaphores)

신호기(semaphores:쎄마포어)는 조작체계의 자원을 경쟁적으로 사용하는 다중프로쎄스에서 행동을 조정하거나 동기화시키는 기술이다. 신호기는 조작체계 또는 핵심부의 지정된 기억장치내의 값으로서 매 프로쎄스는 이것을 확인하고 변경할수있다. 확인되는 신호기의 값에 따라 그 프로쎄스가 즉시 자원을 사용할수도 있으며

또한 이미 다른 프로쎄스에 의해 사용중이라는것을 알게 된 경우에는 재시도하기전에 일정한 시간을 기다려야만 한다. 신호기는 2진수(0 또는 1)를 사용하거나 추가적인 값을 가질수도 있다. 신호기를 사용하는 프로쎄스는 그 값을 확인하고 자원을 사용하는 동안에는 그 값을 변경함으로써 다른 신호기사용자들이 기다리도록 해야 한다.

신호기들은 일반적으로 주기억공간을 공유하거나 파일들을 공유호출하기 위한 두가지의 목적을 위해 사용된다. 신호기들은 프로쎄스간통신(IPC)을 위한 기술중의 하나이다. C언어에서는 신호기들을 관리하기 위한 일련의 대면부 또는 함수들을 제공한다.

Procmail

procmail은 UNIX의 기본적인 우편프로그람인 Sendmail과 련동하여 다양하고 유용한 기능을 제공하는 우편도구인데 설정방법이 매우 복잡하여 그의 사용이 그리 쉽지는 않다. procmail을 사용하면 사용자의 전자우편(E-mail)을 자유로 통제할수 있으며 사용자의 전자우편주소인 《임의의 @사용자의 도메인》중에서《order @사용자의 도메인》으로 수신되는 전자우편을 임의의 다른 전자우편주소들에서 받아보게 하거나 특정한 전자우편주소로 수신되는 우편에 대하여 자동응답기능을 설정할수 있는 등의 작업이 가능하다.

logger

/usr/bin/logger

체계리력의 기록

주어진 통보문을 /var/log/message 파일에 기록한다.

형식 : logger[설정값][통보문...]

-i : 매 행마다 logger의 ....ID를 기록한다.

-s : system log와 함께 통보문을 표준오유장치에 기록한다.

-f file : 지정한 파일명의 내용을 리력에 기록한다.

-p pri : 지정한 pri우선권으로 통보문을 입력한다. pri에는 수자형식으로 오거나《facility.level》형식으로 올수도 있다. 실례를 들어 《-p local3.info》설정값은 local3 facility에서 information1수준으로 통보문이 기록된다. 초기값은 《user.notice.》이다.

-t tag : 지정한 tag를 매 행마다 기록한다.

logger명령문은 주어진 통보문을 보내여 /var/log/message리력에 기록한다. 그러면 아래의 통보문을 리력파일에 보관해보자.

```
# logger System Logger Testing
```

/var/log/message파일의 최신리력부분에서 logger명령으로 보관한 통보문을 볼수 있다.

```
kcc@linux:~
# tail /var/log/messages
Jul  8  03:29:51 ns sshd(pam_unix)[24285]:session closed for user admin
Jul  8  03:29:51 ns 7월 8 03:29:51 su(pam_unix)[24594]: session closed
for user root
Jul  8  03:29:54 ns named[5671]: lame sever resolving '121.155.49.211.in-
addr.arpa' (in '155.49.211.in-addr.arpa'?): 210.117.65.100#53
Jul  8  03:29:54 ns named[5671]: lame server resolving '121.155.49.211.in-
addr.arpa' (in 155.49.211.in-addr.arpa'?): 210.117.65.2#53
```

```
Jul  8  03:29:54 ns named[5671]: lame server resolving '121.155.49.211.in-
addr.arpa' (in 155.49.211n-addr.arpa'?): 210.117.65.100#53
Jul  8  03:29:54 ns named[5671]: lame sever resolving '121.155.49.211.in-
addr.arpa' (in 155.49.211.in-addr.arpa'?): 210.117.65.2#53
Jul  8 03:29:54 ns sshd[24685]: Could not reverse map address 211.49.155.121
Jul  8  03:30:00 ns sshd[24665]: Accepted password for admin from 211.
49.155.121 port 1133 ssh2
Jul 8 03:30:00 ns sshd(pam_unix)[24665]: session opened for user admin
By (uid=0)
Jul 8 07:11:22 ns System Logger Testing
```

【관련명령문】

syslogd : Linux상에서의 체계리력을 기록하는 편의프로그람(utility)

klogd : 핵심부리력 데몬

login

/bin/login

체계접속명령문

체계에 접속하여 개인설정파일을 읽고 작업할수 있는 환경을 갖추기 위한 인증과정명령이다.

형식 : login [설정값] [이름]

-f user : 이미 인증절차를 거친 사용자인 경우에는 지정한 user의 인증절차를 무시하게 된다. 만일 user가 root일 때에는 적용되지 않는다.

-h hostname : 체계접속을 원격접속(telentd rlogind)으로 인식하고 hostname에 체계접속을 시도할 호스트의 이름을 입력한다.

-p : login을 실행할 때 이전의 환경들을 그대로 사용하도록 getty지령에 의해 사용된다.

Linux는 체계를 시동할 때에 init프로그람으로 자신의 설정파일을 읽어들이고 실행에 필요한 모든 getty를 생성하는데 가상조종탁에 사용하기 위한 최소 getty프로그람인 mingetty프로그람을 사용한다. 이것은 체계접속프로쎄스실행을 위한 프로그람으로 말단회선속도를 설정하고 체계접속입력독촉(login prompt)을 실행하는 역할을 한다.

사용자가 접속을 시도하면 mingetty는 login프로그람을 호출하며 login프로그람은 입력독촉상태에서 사용자이름을 받아 암호를 읽게 된다.

인증절차후 제대로 입력되였을 때는 /etc/passwd에 있는 쉘을 받아 /etc/bashrc, /etc/profile 등과 home등록부의 .bash_rc, bash_profile 등의 환경설정파일을 불러와 화면상에 입력독촉기호를 나타내게 한다.

쉘 입력독촉에는 #, $ 등이 있다. #는 뿌리쉘 입력독촉이며 $는 사용자쉘 입력독촉이다. /etc/passwd 에 사용자의 체계접근시도를 막는 방법에는 여러가지가 있는데 해당한 사용자설정의 맨 앞행에 #로 주해처리를 하거나 /bin/bash를 /bin/false 로 변경하는것도 접근을 막을수 있게 하는 한가지 방법으로 된다. 다른 방법으로서 /etc/nologin 파일에서 사용자를 추가하여 접근을 막을수도 있다.

/etc/securetty파일에는 뿌리로 접속할수 있는 말단들이 기입되여있다. 말단이름은 /dev/문자를 생략한 말단이름뿐이다. 이 파일이 없는 경우에 뿌리는 모든 말단에서 접속이 가능하다.

logname

/usr/bin/logname

현재 사용자이름의 보기

현재 체계접속한 사용자이름을 utmp에서 찾아 보여주는 명령문이다.

형식 : logname [설정값]

—help : 사용법을 보여준다.
—version : 판본정보를 보여준다.

logname명령문은 /var/run/utmp파일에 현재 체계접속한 사용자이름을 등록하며 이 파일의 정보를 보여준다. logname과 whoami, id명령문간에는 약간의 차이가 있다. 매 명령문들에서는 체계접속한 사용자를 알수 있다.

```
kcc@linux:~
$ logname
admin
$ whoami
admin
$ id
uid=500(admin) gid=500(admin) groups=500(admin)
```

id명령문을 사용하면 사용자의 UID와 GID, 속한 집단들을 다 볼수 있다. su명령을 사용하여 고급사용자(super user)로 체계접속하여 보자.

```
kcc@linux:~
$ su
Password:
```

logname은 su명령으로 전환된 고급사용자가 아니라 초기 체계접속한 사용자의 이름을 보여준다. 그러나 whoami명령으로 su명령으로 전환된 뿌리를 보여준다.

```
kcc@linux:~

# logname
admin
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
```

【관련명령문】

whoami : 체계접속한 사용자의 ID를 보여주는 명령문

id : 체계접속한 사용자의 UID, GID, groups 를 보여주는 명령문

logrotate

/usr/bin/logrotate

리력파일관리

리력파일을 주기적으로 압축하고 이름을 바꾸어 관리하는 명령문이다.

형식 : logrotate[설정값]<설정파일>

-d : 오유수정방식(debugger mode)

-f, —force : 강제적으로 실행한다. 이 설정값은 새로운 목록이 추가된 경우나 오래된 리력파일을 수동으로 직접 삭제한 경우에 유용하다.

-s, —state [statefile] : 지정된 방책파일을 사용하여 logrotate를 실행한다.

—usage : 간단한 사용법을 보여준다.

cron은 예비보관과 같이 주기적으로 실행하는 명령들을 자동적으로 처리하도록 해주는 도구로서 설정내용은 /etc/crontab에 있다. logrotate는 리력파일(log file)들을 주기적으로 관리하는 도구이다. 지정된 주기에 맞추어 매 리력파일들의 이름바꾸기, 압축, 삭제 등을 진행한다.

보통 /etc/cron.daily에 logrotate라는 스크립트를 두어 /usr/sbin/logrotate를 /etc/logrotate.conf라는 설정에 맞추어 매일 실행하게 되여 있다.

/etc/cron.daily/logrotate 파일을 한번 보자.

```
kcc@linux:~
# cat /etc/cron.daily/logrotate
#!/bin/sh
/usr/sbin/logrotate /etc/logrotate.conf
```

/etc/cron.daily/logrotate파일은 /etc/logrotate.conf파일을 참조하여 실행한다.

그러면 이 파일을 한번 보도록 하자.

```
kcc@linux:~
# cat /etc/logrotate.conf
weekly #1주일에 한번씩 리력파일이 순환된다.
rotate 4 # 최대 4개까지의 예비보관 리력파일을 남긴다.
creat # 4개의 리력파일이 생성된 다음 새로운 리력파일을 생성한다.
compress # 압축으로 리력파일을 보관한다.
include /etc/logrotate.d # rpm패키지리력순환정보가 있는 등록부
/var/log/wtmp { # /var/log/wtmp리력파일에 대한 설정이다.
   Monthly       # 한달에 한번씩 리력파일이 순환된다.
create 0664 root utmp #0664는 생성하는 파일의 사용허가권, root는 소유자 utmp는 집단
   rotate 1     # 예비보관리력이 없이 하나의 리력파일만을 생성한다.
                }
```

우의 설정에서 rpm패키쥐리력순환정보가 있는 등록부를 보자.
매 리력파일설정을 보면 개개의 정보를 볼수 있다.

```
kcc@linux:~
# ls /etc/logrotate.d/
apache   mars-nwe.log mysqld psacct  samba slrnpull  syslog  vm
linuxconf  mgetty  named  rpm  sendfax  squid   vgetty
```

우의 파일중에서 syslog파일을 한번 보자. syslog파일에는 messages나 secure maillog와 같은 중요한 리력파일에 대한 순환정보를 담고있다.

```
kcc@linux:~
# cat /etc/logrotate.d/syslog
/var/log/messages /var/log/secure /var/log/maillog
/var/log/splloer /var/log/boot.log /var/log/cron {
    sharedsripts
    postrotate
     /bin/kill -HUP cat /var/run/syslogd.pid 2> /dev/null 2>
     /dev/null || frue
    endscript
}
```

look

/usr/bin/look

특정문자렬로 시작하는 행을 보기

파일에서 주어진 문자렬로 시작하는 행만 검색하여 보여주는 명령문이다.

형식 : look [설정값] 문자렬 [파일]

-d : 영문자만을 검색한다.

-f : 대소문자를 구별하지 않는다.

-a : 비교사전인 /usr/dict/web2파일을 사용한다.

-t character : 문자렬의 마지막문자를 지정한다.

look명령은 파일에서 검색한 문자렬로 시작하는 행을 찾아 보여준다. 2진파일검색인 경우에는 파일안의 매 행이 먼저 정렬되여야 한다. 파일이 지정되지 않으면 /usr/dict/words파일을 사용한다. 이 파일에는 영문자모단어만 있으며 대소문자는 구별하지 않는다.

lpc

/usr/bin/lpc

인쇄기관리자

행인쇄기를 조종하는 프로그램이다.

형식 : lpc[명령문]

? : 도움말을 보여준다.
active [pr@[host]]
abort {all | printer} : 현재의 인쇄기데몬을 중지하고 지정한 인쇄기의 인쇄를 비활성화한다.
defaultq : LPC프로그람의 기본대기렬(queue)를 보여준다.
defaults : 설정된 기본값을 보여준다.
disable { all | printer} : 지정한 입출력완충대기렬(spool queue)을 비활성화한다.
Down { all | printer} : 모든 인쇄기의 대기렬과 인쇄, 또는 지정한 인쇄기를 비활성화한다.
enable {all | printer} : 모든 인쇄기의 입출력완충대기렬 또는 지정한 인쇄기를 활성화한다.
exit, quit : LPC프로그람을 탈퇴한다.
help : 간단한 도움말을 보여준다.
lpq printer [설정값] : lpc프로그람에서 lpq를 실행한다.
lprm printer jobid [jobid]* : lpc프로그람에서 lprm을 실행한다.
reread [printer [@host]] : LPD봉사기가 인쇄기설정과 printcap 정보를 다시 읽도록 한다.
start {all | printer} : 인쇄대기렬을 활성화하고 모든 인쇄기나 지정한 인쇄기데몬을 시작한다.
status {all | printer} : 모든 인쇄기나 지정한 인쇄기의 상태를 보여준다.
stop {all | printer} : 현재 진행중인 작업을 완료한 후에 지정한 인쇄기데몬을 비활성화한다.
topq printer [jobid] : 순차적작업목록에 지정한 작업을 인쇄대기렬의 맨 우에 준다.
up {all | printer} : 인쇄대기렬을 활성화하고 지정한 인쇄기데몬을 재기동한다.

lpc는 인쇄기를 조종하는 명령문이다. 인쇄기의 상태를 볼수 있고 인쇄기 또는 인쇄대기렬을 활성화 또는 비활성화하며 인쇄대기렬의 작업순차를 변경시킬수 있다. 보통 lpc는 한 대의 Linux콤퓨터에 여러대의 인쇄기가 련결되여있는 경우에 많이 사용된다.

```
kcc@linux:~

# lpc
lpc> status
Printer Printing Spooling Jobs Server Subserver Redirect Status/(Debug)
lp@ns enabled enabled 0   none  none
lpc>
```

lpc> 의 입력방식에서 명령문을 입력하는 호상대화방식으로 인쇄기를 조종할수 있다. 또한 호상대화방식이 아닌 명령행에서도 조종할수 있다.

```
kcc@linux:~

# lpc status
Printer Printing Spooling Jobs Server Subserver Redirect Status/(Debug)
lp@ns enabled enabled 0   none  none
```

【관련명령문】

lpd : 인쇄기데몬
lpq : 기본 인쇄대기렬의 내용을 보여준다.
lpr : 파일내용을 출력한다.
lprm : 인쇄대기렬의 작업을 삭제한다.
lpstat : LP 인쇄기의 정보를 보여준다.

lpq

/usr/bin/lpq

인쇄대기렬의 보기

현재 인쇄대기중인 내용을 보는 명령문이다.

형식 : lpq[설정값][사용자]

-a : printcap에 있는 자료기지의 모든 인쇄기정보를 보여준다.
-l : 긴 형식으로 상세한 정보를 보여준다.
-P printer : 기본인쇄기를 사용하지 않고 지정한 인쇄기를 사용한다.
-V : 인쇄기판본정보를 보여준다.
Num : 지정한 작업번호의 상태를 보여준다.

lpq명령은 lpd데몬을 실행한 다음 인쇄기설정이 제대로 되여있는가를 확인할 때에 유용하게 쓰인다. 설정값을 첨부함이 없이 실행시키면 기본적으로 설정된 인쇄대기렬의 내용을 보여준다.

```
kcc@linux:~

# lpq
Printer: lp@ns (dest lp@61.40.233.14)
  Queue: no printable jobs in queue
  Server: no server active
  Rank     Owner/D      Class Job Files               Size   Time
done      root          A 221 /root/.kde/tmp-ns.li  29065 18:02:50
JetDirect lpd: no jobs queued on this port
```

lpq명령의 내용중에서 Rank항목은 여러개의 인쇄작업중 현재 진행되는 작업에 대한 내용을 보여준다. Rank의 내용이 active인 경우에는 인쇄기가 인쇄 혹은 사용중임을 나타낸다.

Owner항목은 인쇄작업의 소유자를 나타낸다. 그리고 Job항목은 실행중인

인쇄의 작업번호를 보여주는데 해당한 인쇄작업을 lprm명령으로 삭제하려면 이 작업번호를 지정해주어야 한다.

Lpq항목은 크기, 해당 작업이 실행된 시간 등의 정보를 보여준다.

```
kcc@linux:~
# lpq
Printer 'lp@localhost' - cannot open connection -련결이 거부됨
Make sure the remote host supports the LPD protocol
```

lpq명령으로 확인한 결과, LPD련결이 거부된 경우에 lpd데몬이 기동되지 않는 경우가 대부분이므로 이때에는 데몬을 기동시키면 문제를 해결할수 있다.

【관련명령문】

lpc : 행인쇄기를 조종한다.

lpd : 인쇄기데몬

lpr : 파일내용을 출력한다.

lprm : 인쇄대기렬의 작업을 삭제한다.

lpstat : LP 인쇄기의 정보를 보여준다.

lpr

/usr/bin/lpr

인쇄명령

파일내용을 인쇄하는 명령문이다.

형식 : lpr[설정값]파일

-A : AUTH환경변수에 의해 지정된 인증을 리용한다.
-B : printcap에 지정된 추출기(filter)로 인쇄파일의 추출(filtering)을 하지 않는다.
-C class : lpq지령시에 보여주는 class를 지정한다. 이것은 A로부터 Z까지의 우선순위를 가지는데 기본값은 A클라스로서 A는 우선순위가 가장 낮으며 Z는 우선순위가 가장 높다.
-h : banner 또는 header를 포함하지 않는다.
-i indentcols : 지정한 indentcols수자만큼 들여쓰기를 한다. 모든 인쇄기에 지원되지는 않는다.
-J jobname : 지정한 jobname이 작업이름으로 된다. 지정하지 않으면 파일이름이 인쇄대기렬의 작업이름으로 된다.
-Kcopies, -#copies : 지정한 copies수자만큼 파일이 인쇄된다.
-m mailTo : 인쇄작업이 완료된 경우 지정한 mailTo에 우편을 보낸다. 형식은 user@host의 형식을 따르면 된다.
-P printer : PRINTER환경변수나 체계에서 지정된 인쇄기가 아니라 지정한 printer에서 인쇄한다.
-r : spooling작업이 완료된 경우에 파일을 제거한다.
-T title : 지정한 title을 pr에 대한 파일이름으로가 아니라 제목으로 사용한다.
-U username : 지정한 username으로 작업의 사용자를 지정한다. 이러한 사용자로서는 root나 설정파일에서 허용한 allow_user_setting목록에 있는 사용자만이 가능하다.
-V : 진행되는 상황을 상세히 보여준다.
-w width : 지정한 width로 인쇄하는 페지의 너비를 지정한다.

lpr는 지정한 파일을 인쇄할수 있게 하는 명령이다.

파일인쇄의 다른 한가지 방법으로서 아래의 실례와 같이 cat명령으로 파일을 /dev/lp0에 보내여 인쇄하는것이 가능하다. 그러나 이 방법은 뿌리식별이름이나 print집단만이 사용할수 있다.

```
# cat test.txt > /dev/lp0
```

아래의 실례를 보자

col -b명령은 문서페지(man page)의 troff으로 전처리된 형식을 제거한 후, 이 파일을 보기 쉽도록 인쇄하게 한다.

```
$ man man | col -b | lpr
```

-C설정값은 A로부터 Z까지의 우선권순위클라스를 부여할수 있다. 기본값은 A클라스로써 우선권순위가 가장 낮으며 Z는 우선권순위가 가장 높다. lpr명령으로 test.txt 파일을 인쇄할 때 이 파일에 B의 우선권순위를 부여할 수 있다.

```
$ lpr -C B test.txt
```

-k2이나 -#2설정값으로 해당한 파일을 지정하는 수자만큼 출력할수 있다.

```
$ lpr -#2 test.txt
```

참 고

troff

UNIX조작체계에서 사용하는 영문문자인식프로그람으로서 타자기와 같이 문자폭이 일정한 출력장치용으로 만들어진 nroff(new roff)를 사진문자인식용으로 바꾸어 쓴것이다. troff는 본래 사진문자인식기였지만 현재는 출력기로서 레이자인쇄기만을 사용한다.

lprm

/usr/bin/lprm

인쇄취소

인쇄대기렬의 작업을 삭제하는 명령문이다.

형식 : lprm[설정값][작업번호][사용자]

-P printer : 지정한 printer에 해당되는 작업들만을 보여준다.
- : spool 사용자가 가지고있는 모든 대기작업을 삭제한다.

lprm은 lpq명령으로 볼수 있는 작업대기렬들을 확인하여보고 해당하는 작업을 취소할수 있다.

작업번호를 지정하여 작업번호에 해당하는 대기렬을 삭제한다. 아래의 례를 보기로 하자.

```
kcc@linux:~

# lpq
Printer: lp@ns (dest lp@61.40.223.14)
 Queue: no printable jobs i queue
 Server: no server active
Rank     Owner/ID   Class  Job  Files              Size   Time
Active   root       A      201  /root/text.txt     52     18:01:20
Waiting  root       A      221  /root/.kde/tmp/ns.li 29065 18:02:50
JetDirect lpd: no jobs queued on this port
```

현재 인쇄중인 201작업번호와 대기렬에 있는 221작업번호가 있다. 이 작업중에서 대기중인 221작업번호를 삭제해보자.

```
# lprm 221
```

확인해보면 이 작업이 대기렬에서 사라졌다는것을 알수 있다.

```
kcc@linux:~

# lpq
Printer: lp@ns (dest lp@61.40.223.14)
 Queue: no printable jobs in queue
JetDirect lpd: no jobs queued on this port
```

lprm사용자를 지정하여 해당하는 사용자의 작업만을 취소할수 있는데 이것은 뿌리만이 사용할수 있다.

```
# lprm admin
```

또한 lprm - 명령으로 spool내의 모든 작업을 취소할수도 있다.

```
# lprm -
```

【관련명령문】

lpc : 행인쇄기를 조종한다.
lpd : 인쇄기데몬
lpq : 인쇄대기렬정보를 보여준다.
lpr : 파일을 인쇄한다.
lpstat : LP 인쇄기의 정보를 보여준다.

ls

/bin/ls

파일목록보기

등록부와 파일의 목록을 보여주는 명령문이다.

형식 : ls [설정값] [파일]

-a —all: 《.》을 포함한 경로안의 모든 파일을 보여준다.

-b, —escape : 자모형식을 사용하는 파일이름안에서 도형문자가 아닌 문자들을 사용한다.

-c, —time=ctime, —time=status : 파일의 최근 변경시간에 따라 정렬하여 보여준다. 자세한 라렬(-l설정값)이면 그 파일의 최근 변경시간을 보여준다.

-d, —derectory : 경로안의 내용을 라렬하지 않고 그 경로를 보여준다(이것은 쉘스크립트에서 유용하게 쓰인다).

-f : 경로내용을 정렬하지 않는다. 이것은 디스크에 보관된 순서로 보여준다. -a, -U설정값과 같은 의미이며 -l, -s, -t, 설정값과는 반대의 의미이다.

—full-time: 시간을 간략형식으로가 아니라 완전한 형식으로 보여준다.

-i, —inode : 파일왼쪽에 색인번호를 보여준다.

-k, —kilobytes : 파일크기가 라렬되면 KB단위로 보여준다.

이 설정값은 POSIXLY_CORRECT환경변수를 무시한다.

-l, —format=long, —format—verbose : 파일을 라렬할 때에 파일형식, 사용권한, 하드련결(Hard link)번호, owner이름, group이름, 파일크기, 시간(따로 지정하지 않으면 파일이 만들어진 날자) 등을 자세히 보여준다. 시간은 6개월이전의것인 경우에 시간이 생략되고 파일의 년도가 포함된다.

-m, —format=commas : 파일을 가로형식으로 라렬한다. 가로형식으로 라렬가능한만큼 최대로 라렬하여 보여준다.

-n, —numeric-uid-gid : 이름의 라렬에서 UID, GID번호를 사용한다.

-p : 파일형식을 지시하는 문자를 매 파일에 추가한다.

-q, —hide-control-chars : 파일이름에 도형문자가 아닌것이 있으면 ?' 로 표시한다.

r, -reverse : 정렬순서를 내림순차로 한다.

-s, —size : 파일크기를 1KB단위로 나타낸다.

POSIXLY_CORRECT환경변수가 지정되면 512byte단위로 정해진다.

-t, —sort=time : 파일시간순차로 정렬된다. 최근의 파일을 제일 먼저 보여준다.

-u, —time=atime, —time=access, —time=use : 파일사용시간순차로 정렬한다. 자세한 라렬인 경우, 시간표시는 만들어진 날자대신에 사용된 날자를 보여준다.

-x, —format=across, —format=horizontal: 정렬방식을 가로로 한다.

-A, —almost-all : 《.》, 《..》경로를 제외한 등록부안의 모든 파일을 라렬한다.

-B, —ignore-backups : 파일 끝이 ~인 파일은 목록라렬에서 제외된다.

-C, —format=vertical : 정렬방식을 세로로 한다.

-F, —classify : 파일형식을 알리는 문자를 각 파일뒤에 추가한다. 일반적으로 실행파일은 《*》, 경로는 《/》, 기호련결은 《@》, FIFO는 《|》, 소케트는 《=》 등이며 일반적인 파일은 없다.

-G, —no-group : 자세한 목록라렬에서 group정보를 제외한다.

-L, —dereference : 기호련결파일들을 파일 그대로 보여준다.

-N, —literal : 이름이 영문이 아닌 경우, C언어에서 사용하는 역사선문자(\)와 함께 사용하는 표기대신에 그대로 출력한다.

-Q —quote-name : -N설정값과 반대이다.

-R, —recursive : 하위경로와 그 안에 있는 모든 파일들도 라렬한다.

-S, —sort=size : 파일크기가 가장 큰것부터 정렬하여 라렬한다.

-U, -sort=none : 정렬을 하지 않고 디스크에 보관된 순차로 보여준다. 이 설정값은 -f설정값을 사용할수 없다. 그것은 UNIX용의 ls -f인 경우, -a 설정값은 가능하지만 -l, -s, -t설정값이 불가능하기때문이다.

-X, —sort=extension : 파일확장자순차로 정렬한다. 확장자가 없는 파일이 제일먼저 라렬된다.

-1, —format=single-column : 하나의 column에 한 파일씩 라렬한다.

-w, —with cols : 가로길이를 cols값으로 지정한다. 기본적으로 한 화면의 가로값으로 된다. 또한 COLUMNS환경변수값으로 지정할수 있다. 초기값은 80이다.

-T, —tabsize cols : tab가 사용될 때에 cols값으로 지정한다. 초기값은 8이다. 0으로 지정되면 tab문자는 무시된다.

-I, —ignore pattern : pattern으로 지정된 파일들은 목록에서 제외된다. 이때 지령행에서 그 파일이 지정되면 물론 라렬된다.

—color, —colour, —color=yes, —colour=yes : 파일의 형태에 따라 그 파일의 색을 다르게 보여주는 기능을 한다. 자세한 설명에 대해서는 아래의 DISPLAY COLORIZATION 부분을 참조하기 바란다.

—color=tty, —colour=tty : —color설정값과 같은데 다만 표준출력에서만 색을 사용한다. 이 설정값은 색조종코드를 지원하지 않는 쉘스크립트나 지령행사용에서 아주 유용하게 쓰인다.

—color=no, —colour=no : 파일을 구분하는 색을 사용하지 않는다. 이것이 초기값이다. 이 설정값은 색사용을 이미 하고있다면 무시된다.

—help : 도움말을 보여준다.

—version : 판본정보를 보여준다.

ls는 표준출력으로 지정한 등록부나 파일의 정보를 보여준다.
파일이나 등록부를 지정하지 않으면 현재 등록부의 내용을 보여준다.
현재 등록부에서 숨긴파일까지 모두 보려는 경우에는 -a설정값을 사용한다.

```
# ls-a
```

-l설정값은 해당되는 파일의 상세한 정보를 보여준다.

```
kcc@linux:~
# ls -l .bashrc
-rw-r-r-    1root    root    176    8월 24    1995    .bashrc
```

-d설정값은 지정한 등록부의 내용만을 보여준다.

```
kcc@linux:~
# ls -ld /etc /var
drwxr-xr-xr-    70    root    root    8192    7월 9 16:34    /etc
drwxr-xr-xr-    28    root    root    4096    7월 4 21:51    /var
```

-full-time 설정값을 사용하면 자세한 날자정보를 볼수 있다.

```
kcc@linux:~

# ls --full-time
합계 3036
drwxr-xr-x    3    root    root    4096   2003-12-04
11:41:07.000000000 +0900    CORBA
-rw-r--r--    1 root root    2673    2004-01-23
15:39:44.000000000 +0900    DIR_COLORS
drwxr-xr-x    5    root    root    4096   2003-12-14
19:02:15.000000000 +0900    FreeWnn
-rw-r--r--    1 root root    77681 2003-11-05 11:14:30.000000000
+0900          Muttrc
drwxr-xr-x    20    root    root    4096   2004-07-05
00:44:35.000000000 +0900    X11
```

ls명령을 사용하였을 때에 출력되는 파일크기는 블로크단위로서 쉽게 그 크기를 알아보기는 어렵다. -h설정값은 파일크기를 사람이 인식하기 쉬운 단위로 보여준다.

```
kcc@linux:~

# ls -lh
합계 3.0M
drwxr-xr-x   3 root  root   4.0K 12월  4 2003   CORBA
-rw-r-r--   1 root   root   2.6K  1월 23 15:39   DIR_COLORS
drwxr-xr-x    5 root  root   4.0K 12월 14 2003   FreeWnn
-rw-r-r--    1 root   root   76K   11월  5 2003   Muttrc
drwxr-xr-x   20 root   root   4.0K   7월   5  00:44   X11
-rw-r-r--    1 root   root   2.5K 10 월  13 2003  a2ps-site.cfg
```

-r설정값은 일반적인 ls결과의 역순차로 정렬하여 보여준다.

```
kcc@linux:~

# ls -rl
합계 3036
-rw-r-r—    1 root   root      460  10월 12  2003 zshrc
-rw-r-r—    1 root   root      303  10월 12  2003 zshenv
-rw-r-r—    1 root   root      129  10월 12  2003 zprofile
-rw-r-r—    1 root   root       86  10월 12  2003 zlogout
-rw-r-r—    1 root   root      253  10월 12  2003 zlogin
drwxr-x-    2 root   root     4096  12월 14  2003 zshrc
```

-t설정값은 가장 최근에 변경된 파일의 순차로 정렬하여 보여준다.

-r설정값을 추가하여 가장 마지막에 변경된 시간으로부터 출력현시되도록 할 수 있다(-trl).

```
kcc@linux:~

# ls -tl | more
합계 3036
drwxr-xr-r  2 root   root     4096   7월 9 04:46 mail
-rw-r-r—    1 root   root   109882   7월 9 02:51 ld.so.cache
-rw-r-r—    1 root   root      937   7월 8 15:59 named.conf
drwxr-xr-r  2 root   root     4096   7월 8 15:37 samba
drwxr-xr-r  9 root   root     4096   7월 8 03:17 sysconfig
drwxr-xr-r  2 root   root     4096   7월 7 23:58 cron.daily
drwxr-xr-r  2 root   root     4096   7월 8 23:58 cron.weekly
```

-s설정값은 파일크기가 가장 큰것부터 보여준다. 만일 가장 작은것부터 출력되도록 하려면 -r설정값을 추가하여 -Srlh설정값으로 볼수 있다.

```
kcc@linux:~

# ls -Srlh | more
합계 3.0M
-rw-r-r--   1 root   root    720K  10월 13 termcap
-rw-r-r--   1 root   root    132K  11월  3 lynx.cfg
-r-r-r--    1 root   root    127K  11월  3 lynx.cfg.cs
-r-r-r--    1 root   root    127K  11월  3 lynx.cfg.sk
-r-r-r--    1 root   root    127K  11월  3 lynx.cfg.ja
-rw-r-r--   1 root   root    107K   7월  9 ld.so.cache
```

lsattr

/usr/bin/lsattr

파일속성보기

Linux파일체계의 속성을 보여주는 명령문이다.

형식: lsattr[설정값][파일]

-R : 등록부와 그 안의 모든 내용에 대해 속성값을 출력한다.
-a : 《.》으로 시작하는 파일을 포함한 파일등록부안의 모든 파일을 보여준다.
-d : 파일등록부를 보여줄 때 그 안에 든 파일을 보여주는것이 아니라 일반파일과 마찬가지로 보여준다.
-v : 파일판본을 출력한다.

lsattr명령문은 chattr명령으로 변경한 파일의 속성내용을 볼수 있게 한다(구체적인 사용법에 대해서는 chattr명령을 참고할것).

chattr +i설정값으로 /etc/passwd에 i속성을 부여해보자. 이 속성은 파일의 삭제, 이름변경, 내용추가, 련결의 생성 등을 모두 할수 없게 한다.

```
kcc@linux:~
# chattr +i /etc/passwd
```

chatttr명령으로 변경된 속성을 보기 위해서는 lsattr명령이 필요하다.

```
kcc@linux:~
# lsattr /etc/passwd
----i------ /etc/passwd
```

【관련명령문】

chattr : 파일체계의 파일속성을 변경하는 명령문

lsdev

/sbin/lsdev

하드웨어정보의 보기

하드웨어의 장치정보와 현재 상태를 보여준다.

형식 : lsdev

lsdev명령은 체계에 인식된 하드에어의 장치명과 DMA, IRQ, I/O포구 등의 정보를 보여주는 명령문이다. 이 명령은 체계에 인식된 하드웨어정보를 한 눈에 볼수 있게 해주는 우점이 있다.

```
kcc@linux:~

# lsdev
device      DMA   IRQ   I/O    Ports
------------------------------------------------------------
8139too                 e800-e8ff ec00-ecff
cascade      4     2
dma                     0080-008f
dma1                    0000-001f
dma2                    00c0-00df
eth0              10
eth1               5
fpu                     00f0-00ff
ide0              14    01f0-01f7 03f6-03f6 e000-e007
ide1              15    0170-0177 0376-0376 e000-e00f
keyboard          1     0060-006f
Mouse             12
PCI                     0cf8-0cff
pic1                    0020-003f
pic2                    00a0-00bf
Realtek                 e800-e8ff ec00-ecff
rtc                8    0070-007f
timer              0    0040-005f
usb-uhci          11    e400-e41f
vga+                    03c0-03df
VIA                     e000-e00f e400-e41f
ymfpci             7
```

lsmod

/sbin/lsmod

적재된 모듈의 보기

현재 동작하고있는 모듈들을 보여주는 명령문이다.

형식 : lsmod

lsmod명령으로는 체계에서 현재 동작하고있는 모듈들을 확인할수 있다.

```
kcc@linux:~

# lsmod
Moudle          Size     Used by    Tainted: P
smbfs           35552    2          (autoclean)
ymfpci          42660    0          (autoclean)
uart401         6560     0          (autoclean) [ymfpci]
sound           59052    0          (autoclean) [uart401]
ac97_codec      9504     0          (autoclean) [ymfpci]
soundcore       4324     4          (autoclean) [ymfpci sound]
autofs          10948    0          (autoclean) (unused)
8139too         17440    2
ipx             16404    0          (autoclean)
ipchains        37704    0
ext3            61600    2          (autoclean)
jbd             40452    2          (autoclean) [ext3]
usb-unci        21764    0          (unused)
usbcore         51744    1          [usb-uhci]
```

보다 자세한 설명에 대해서는 modprobe명령문을 참고하기 바란다.

【관련명령문】

insmode : 체계에 모듈을 적재하는 명령문

modprbe : insmod보다 지능적인 체계모듈적재명령문

lsof

/usr/bin/lsof

사용중인 파일들의 목록보기

현재 실행하고있는 파일의 목록을 확인하고 파일정보와 프로세스정보를 보여준다.

형식 : lsof[설정값][파일]

-a : 파일을 선택하는데서 AND연산에 의해 대상을 보여준다.
-c : 지정한 COMMAND항목의 내용만 보여준다.
-F : 지정한 식별자로 항목(field)을 구분하여 보여준다.
-g : 지정한 집단ID를 사용하는 관련 프로세스를 보여준다.
-I : 현재 사용되는 소케트정보를 보여준다.
-l : login이름대신에 UID를 출력한다.
-n : 호스트이름대신에 IP주소를 보여준다.
-P : 포구봉사이름대신에 포구번호를 보여준다.
-r : 지정한 초의 주기로 반복해서 보여준다. 기본값은 15초이다.
-s : 파일크기를 보여준다.
-V : 자세한 정보를 보여준다.
-?, h : 도움말을 보여준다.

lsof명령문은 현재 열려 있는 모든 파일들에 대한 자세한 표와 매 파일에 대한 소유자, 크기, 관련프로세스 등의 정보를 보여준다.

```
kcc@linux:~

# lsof
COMMAND   PID  USER     FD   TYPE   DEVICE   SIZE    NODE NAME
init        1  rootmem  REG  3,5    30748    30303   /sbin/init
init        1  rootmem  REG  3,5    73120    8069    /lib/ld-2.1.3.so
init        1  rootmem  REG  3,5    931668   8075   /lib/libc-2.1.3.so
cardmgr   213  rootmem  REG  3,5    36956    30357   /sbin/cardmgr
…
```

COMMAND항목중 sshd와 관련된 체계에서 사용하는 파일들을 보자.

```
kcc@linux:~

# lsof
COMMAND  PID   USER  FD   TYPE  DEVICE  SIZE    NODE NAME
sshd     799   root  cwd  DIR   3,8     4096    2  /
sshd     799   root  rtd  DIR   3,8     4096    2  /
sshd     799   root  txt  REG   3,8     258028 343237 /usr/sbin/sshd
sshd     799   root  mem  REG   3,8     464005 228056 /lib/ld-2.2.4.so
sshd     799   root  mem  REG   3,8     35424 228220 /lib/libpam.so.0.75
sshd     799   root  mem  REG   3,8     65353 228069 /lib/libdl-2.2.4.so
sshd     799   root  mem  REG   3,8     47504 228109 /lib/libutil-2.2.4.so
sshd     799   root  mem  REG   3,8     59618 609628 /usr/lib/libz.si.1.1.3
sshd     799   root  mem  REG   3,8     448285 228074 /lib/libnsl-2.2.4.so
```

-i설정값을 사용하여 현재 열려있는 소케트정보를 볼수 있다. 또한 TCP와 UDP를 각각 지정하여 볼수도 있다. 체계에서 NAME항목의 '*:봉사명'에 의해 현재 봉사대기상태를 아래의 례와 같이 확인할수 있다.

```
kcc@linux:~

# lsof -iTCP
COMMAND    PID   USER   FD   TYPE DEVICE   SIZE NODE NAME
sshd       799   root   3u   IPv4  1407        TCP *:ssh (LISTEN)
lpd        817   root   6u   IPv4  1425        TCP *:printer (LISTEN)
X         1917   root   1u   IPv4  12765       TCP *:x11 (LISTEN)
rdesktop 2607   root   7u   IP4v  18494      TCP ns.linuxroot.co.kp:1530-
>61.40.233.213:3389 (ESTABLISHED)
xinetd      2691   root   3u   IPv3   20721     TCP *:pop3 (LISTEN)
sendmail     4420 root  4u   IPv4   26889      TCP *:smtp (LISTEN)
httpd      17897   root   19u  IPv4   75819      TCP *:http (LISTEN)
-------------------------중간 생략-------------------------
sshd        26359 root     4u  IPv4  132664      TCP ns.linuxroot.co.kp.:ssh-
>211.49.155.121:1274 (ESTABLISHED)
sshd        26690 root     4u  IPv4  199030      TCP ns.linuxroot.co.kp.:ssh-
>211.49.155.121:1064 (ESTABLISHED)
sshd        26792 root     4u  IPv4  199121      TCP ns.linuxroot.co.kp.:ssh-
>211.49.155.121:1031 (ESTABLISHED)
```

M

mail /bin/mail

우편의 송수신

말단에서 사용하는 우편의뢰기프로그람이다.

형식 : mail[-설정값][-s 제목][-c 참조][-b 숨은 참조]받는 사람
mail[-설정값] -f 파일이름
mail[-설정값] -u 사용자 ID

-v : 우편을 보내는 과정의 통보문을 자세히 보여준다.
-i : tty새치기신호를 무시한다.
-n : 우편프로그람의 시작때에 /etc/mail.rc 파일을 읽지 않는다.
-N : 초기화면에서 우편목록을 보여주지 않는다.
-s 제목 : 발송우편의 제목을 적는다.
-c 참조 : 함께 우편을 받을 사람을 추가한다.
-b 숨은참조 : 함께 우편을 받는 사람을 추가한다. 숨은 참조된 사람은 우편을 받은 다른 사람에게는 보이지 않는다.
-f : 파일로부터 우편을 읽어 온다.
-u : 식별이름지정

mail은 말단에서 사용하는 우편의뢰기프로그람으로서 우편의 발송, 받기, 정리 등을 할수 있다.

받은 우편의 확인

자기의 식별이름을 사용중에 받은 우편을 확인하려면 mail만을 입력한다.

관리자방식에서 특정 식별이름의 받은 우편을 확인할 때의 사용실례를 아래에 주었다.

```
kcc@linux:~

# mail -u ebook
Mail version 8.1 6/6/93. Type ? for help
"/var/mail/ebook": 5 message 3 new
> 1 first@kccsinhung.com Tue Jul 6 14:30 62/2080 "[광고]최신 콤퓨터부분
품의 판매봉사"
> 2 ckm@kccmail.net Tue Jul 5 03:23 62/2080  " "
> N 3 ebook@kcctongbo.com Tue Jul 2 03:30 154/580 "안녕하십니까?"
> N 5 kbs@kccbongsa.com Tue Jul 1 08:30 143/3512 "[모든]CD를 판매합
니다."
&
```

mail을 실행하면 초기화면에 받은 우편의 목록이 나온다. 초기화면에 나오는 부분을 우편의 머리부분이라고 하는데 다음과 같이 구성되여있다.

[주의]

N/P/U란? 파일이나 서류철에로의 보관여부를 나타낸다.

N/P/U/	5	ebook@ kcctongbo.com	Tue Jul 1 08:30	143/3512	《안녕하십니까?》
우편상태	페지번호	보낸 사람	날자 및 시간	행/문자수	우편제목

우편 상태

N	새 통보문
P	체계의 우편함(mail box)에 보관
U	읽지 않은 우편

처음 수신된 우편부터 확인하려면 [Enter]건을 누른다. 특정한 우편을 보려고 할 때에는 &우편번호를 지정한다.

```
kcc@linux:~
&1 〈Enter〉
Message 1:
From root Tue Jul 6 13:18:41 2004
Date: Tue, 6 Jul 2004 13:18:41 +0900
From: LinuxConx kbs@kccbongsa.com
To: ebook@kcctongbo.com
Subject: 시험우편을 보냅니다.
시험우편입니다.
&
```

mail명령문

mail의 여러 기능은 명령문을 통해 이루어진다. 명령문입력방식은 《&》로 표시한다.

명령문	설 명
n	다음 우편으로 이동한다.
F〈우편번호〉	우편목록을 보여준다. f* 또는 & f a 모든 우편목록을 볼수 있다.
d〈우편번호〉	지정한 우편번호의 우편을 지운다.
s〈우편번호〉	파일에 지정한 우편내용을 보관한다.
u〈우편번호〉	우편을 지우지 않는다.
R〈우편번호〉	회신한다.
r〈우편번호〉	전체를 회신한다.
q	mail완료
x	mail완료, 체계우편함에서 삭제하지 않는다.
h	통보문의 머리부분을 출력한다.
!	쉘명령문을 실행한다.
e	Ex방식으로 통보문을 편집한다.

회신을 보내기

우편보기명령창에서 현재우편에 대한 회신을 보내려고 할 때에는 R명령을 입력한다. 내용을 다 입력하면 [Ctrl+D] 건을 눌러 우편입력을 완료한다.

[주의]

참조가 없는 경우에 그냥 [Enter] 건을 누른다.

```
kcc@linux:~

To: ebook@kcctongbo.com
Subjecu: Re: 시험우편을 보냅니다.
회신우편입니다.

<Ctrl+D>
CC:<Enter>
```

우편보내기

mail을 실행한 상태에서 명령창에 &m주소의 형식으로 입력하는것만으로도 우편을 발송할수 있다.

Mail을 실행시키지 않은 상태에서 참조(-c), 숨은 참조(-b)를 포함한 우편을 발송해보자. 우편이 보내는 과정을 보려면 -v설정값을 사용한다.

[주의]

우편을 쓰다가 내용을 수정하려고 할 때에는 새로운 행의 첫번째 칸에 ~v라고 입력하면 vi편집기상태에서 내용을 수정할수 있다.

```
kcc@linux:~

# mail -v -c ckm@kccmail.net -b kbs@kccbongsa.com ebook@
kctongbo.com
Subjecrt:[Test]우편발송시험중입니다.
안녕하십니까? 어떻게 지내시는지요.
오래간만에 편지를 보내게 됩니다.

<Ctrl+D>
CC: ckm@kccmail.net <Enter>
ebook@kcctongbo.com... Connecting to mailkcccom.com. via esmtp...
220 mail.kcc.com.ESMTP Sendmail 8.12.5/8.12.5; Sun, 14Jul 2004
12:23:19 + 0900
>>> HELO I
>>> QUIT
---------------------중간 생략---------------------------
ebook@kcctongbo.com… Sent (g6E3NJIn020985 Message accepted
for delivery)
closing connection to mail.kcc.com
```

[주의]

본문중에 본문파일을 첨부할 때에는 세번째 행의 첫번째 칸에 ~r파일이름형식으로 입력한다.

mail명령으로는 2진파일을 첨부할수 없었으므로 파일을 복호화하여 본문형태로 전송할수 있게 만들어 보내군 하였다.

현재는 pine, Netscape Mailer 등 파일첨부가 자유로운 우편의뢰기가 많이 사용되고있다.

mailq

/usr/bin/mailq

우편작업보기

발송될 우편의 정보를 본다.

형식 : mailq[설정값]

-v : 대기렬의 내용을 자세히 보여준다.

sendmail을 통해 발송되는 우편은 대기렬(queue)에 쌓여져 발송대기를 하게 되는데 mailq명령문은 우편이 대기되여있는 목록을 보여준다.

```
kcc@linux:~

# mail -v
            /var/spool/mqueue (3 requests)
—Q-ID— —Size— -Priority- —Q-Time— —Sender/Recipient——
g68ISSb02248      4       47 Jul 9 03:28 root
             (host map: lookup (ylug.or.kp): deferred)
                                  ebook@ylug.or.kp
g693www02498    6       130 Jul 9 12:58 root
             (host map: lookup (ebook.kcctongbo.com): deferred)
```

QD	우편의 대기렬 ID를 보여준다.
Size	본문의 크기를 보여준다.
Priority	우편의 우선권순위를 보여준다.
Q-Time	우편이 대기렬에 들어와 대기하기 시작한 시간을 보여준다.
Sender/Recipient	우편의 발신인과 수신인을 보여준다. 또한 우편의 상태를 볼수 있다. 우에서는 deferred로 씌여지고 발송되여 우편이 연기되였음을 알수 있다.

【관련명령문】

sendmail-bp : 발송될 우편이 대기한 상태를 보여준다. Mailq와 같은 명령문이다.

man

/usr/bin/man

문서보기

알려고 하는 명령문의 안내설명문(manual)문서페지를 보여준다.

형식 : man[-설정값][section][-M path][-P pager] [-S list][-m system][-p string]명령문

-a : 찾으려고 하는 명령문이 검색된 문서페지를 모두 보여준다.
-c : 최신의 cat페지가 있어도 원천문서페지를 다시 구성한다.
-d : 실지의 문서를 보여주지 않고 오유수정정보구성을 보여준다.
-D : -d설정값의 구성을 보여준 후에 문서를 보여준다.
-f : whatis명령과 동일하다.
-h : 도움말을 보여준다.
-k : apropos명령과 동일하다.
-K : 모든 문서페지에서 지정한 문자를 찾는다.
-t : /usr/bin/groff으로 페지형식을 보여준다.
-w : 찾으려고 하는 문자의 문서페지가 있는 위치를 보여준다.
-W: -w와 비슷한데 추가정보없이 한행에 하나씩 표시한다.
-C **파일명** : 문서페지의 configure파일을 지정한다. 기본은/etc/man.config파일이다.
-M path : 문서페지검색을 위한 path를 지정한다.
-P **페지** : 지정한 pager로 페지를 지정한다.
-S **목록** : 두점(:)으로 구분한 대화목록(session list)
-m system : 주어진 체계이름에 의해서 검색되는 문서페지를 지정할수 있다.
-p string : nroff or troff 의 앞에서 실행되는 전처리기의 차례를 지정한다.

명령문의 의미나 사용법을 모르는 경우에 사용한다. man파일들을 검색하여 해당 명령문의 안내문서를 보여준다.

```
kcc@linux:~

# man httpd
---------------------------------------------------------------
httpd(8)                                     httpd(8)
NAME
     httpd - Apache hypertext transfer protocol server
SYNOPSIS
     httpd [-x ][-R libexecdir ][ -d serverroot ][ -f config ][
     -C directive][ -c derective][-D parameter ]
     httpd[-h][-l][-L][-v][-V][-S][-t][-T]
DECRIPTION
     httpd is the Apache HyperText Transfer Protocol (HTtP)
     server program, It is designed to be run as a standalone
     deamon process, When used like this it will create a pool
     of child processes to handle requests. To stop it, send a
     TERM signal to the initial(parent) process. The PID of this
     process is written to a file as given in the internet daemon
     inetd(8) each time a connection to the HTTP service is
     made.

     This manual page only lists the command line arguments. For
     details of the derectives necessary to configure httpd see the
```

만일 지정한 명령문을 포함하고있는 문서페지를 모두 보려고 하는 경우에는 -K설정값으로 찾아볼수 있다. -K설정값은 모든 문서를 검색하여 명령문이 포함된 모든 문서를 보여준다. 검색된 명령문은 [ynq]중 하나를 선택하여 문서보기(y), 문서를 보지 않기(n), man명령으로부터 탈퇴(q) 등을 결정할수 있다.

mtab관련파일을 -K설정값으로 보자.

```
kcc@linux:~

# man -K mtab
/usr/share/man/ko/man8/umount.8.gz? [yng] n
/usr/share/man/ko/man8/rdev.8.gz [ynq] n
/usr/share/man/ko/man8/mount.8.gz? [ynq] n
grep: /usr/share/man/man/man1: 등록부입니다
/usr/share/man/man8/ncpmount.8.gz? [ynq]
```

문서페지는 매개의 식별란으로 나누어져 있다. 문서페지의 제일 웃단의 괄호안의 수자나 압축된 문서페지파일이름의 끝수자가 나타내는 의미로서 아래에 주었다.

1	실행프로그람 혹은 쉘명령문
2	체계호출(핵심부제공함수)
3	서고호출(체계서고포함함수)
4	특수파일(대체로 /dev아래의 파일)
5	파일형식집합 (례: /etc/passwd)
6	게임관련
7	마크로패키쥐집합
8	체계관리명령
9	핵심부루틴 (비표준)

우의 -K설정값으로 처음 찾은 umount.8.gz파일은 8의 의미로서 체계관리명령으로 볼수 있다.

【관련명령문】

apropos : 문서페지설명에 특정단어를 포함한 명령문을 보여준다. (man -k)
whatis : 찾으려고 하는 명령문단어를 확장한다.(man -f)

mattrib

/usr/bin/mattrib

MS-DOS파일의 속성변경

MS-DOS파일의 속성을 변경한다.

형식 : mattrib[±속성]DOS파일[DOS파일]

±a : achive비트로서 여벌복사프로그람이 새로운 파일을 지정할 때에 사용된다.
±r : 읽기전용비트로서 읽기전용으로 지정한다.
±s : 체계비트로서 MS-dos가 체계파일을 가리킬 때 사용된다.
±h : 숨김파일비트로서 숨김파일을 만들 때 사용된다.

MS-DOS파일의 속성설정을 변경한다.
속성의 변경은 의도하는 속성의 추가(+)와 제거(-)를 통해 이루어진다.
dosfile.txt파일에 읽기전용과 안보이기속성을 부여하는 실례를 아래에 주었다.

```
kcc@linux:~

# mattrib +rh dosfile.txt

# mdir a:
Volume in drive A is EBOOK
Volume Serial Number is 0341-CC70
Directory for A:/

test              1093 07-06-2004 19:34 test
        1 file              1 093  bytes
                         1  454 592 bytes free
```

mdir명령의 모두 보기속성(-a)으로 숨긴파일을 확인할수 있다.

```
kcc@linux:~

# mdir -a a:
Volume in drive A is EBOOK
 Volume Serial Number is 0341-CC70
Directory for A:/
test  1093 07-06-2004 19:34 test
dosfile.txt1093 07-06-2004 19:34 dosfiles.txt
             2 file 2 186 bytes
      1 454 592 bytes free

# mdel dosfile.txt
medel: "dosfile.txt" is read only, erase anyway(y/n)? n
```

읽기전용파일을 삭제할것인가를 물어본다.

참 고

bin CD image에서 파일풀기

doldol.bin이라는 image를 Linux에서 바로 사용해보자. 일단 bchunk라는 편의프로그람을 구한 다음에 아래와 같이 실행한다.

```
# bchunk foobar.bin foobar.cue foobar
```

bin2iso를 리용하는 경우에는 다음과 같이 입력한다.

```
# bin2iso doldol.cue
```

이 명령은 bin파일을 iso파일로 풀어낸다. 이것을 적당한 등록부에 마운트하면 CD-ROM을 사용하는것과 같이 리용할수 있다.

```
# mount -t iso9660 -o loop foobar.iso /mnt/cdrom
```

오유가 발생하는것은 대체로 loop device가 없는것이므로 핵심부를 번역할 때 추가해주도록 한다.

mbadblocks

/usr/bin/mbadblocks

MS-DOS파일체계의 불량블로크 표시

MS-DOS파일체계로 초기화한 디스크의 불량블로크를 검사한다.

형식: mbadblocks 구동기:

MS-DOS파일체계를 가지고있는 플로피디스크의 불량블로크를 검사한다. 플로피디스크를 MS-DOS파일체계인 FAT형태로 초기화하고 불량블로크를 검사할 때에 쓰인다.

```
kcc@linux:~

# mbabloocks a:
plain_io : Input/output error
Bad cluster 1901 found
Plain_io : Input/output error
Bad cluster 1902 found
```

1901, 1902의 클라스터(cluster)에 문제가 있다는것을 보여준다.

참　고

FAT파일체계(MS-DOS파일체계)

파일체계는 조작체계가 파일을 보관하고 읽어들이는 규칙이다. 모든 조작체계들에서는 그에 독특한 파일체계를 만들어 조작하는것이 보통이다. 물론 필요에 따라 서로 다른 조작체계의 파일체계가 호환성을 가지는 경우도 있다.

Windows계렬에서는 FAT라고 불리우는 파일체계를 사용한다. 원래는 FAT16을 사용하였는데 Windows 95 OSR-2, Windows 98에서는 성능이 훨씬 개선된 FAT32가 사용되게 되였다.

mcd

/usr/bin/mcd

MS-DOS등록부변경

현재 사용하는 등록부를 변경한다.

형식 : mcd[DOS 등록부]

mtools에 포함된 명령문은 명령문인수부분의 drive를 DOS에서와 같이 A:C:D: 등으로 사용가능하다. MS-DOS등록부로 지정된 내용은 mcd명령문으로 확인할수 있다. 또한 현재 사용하는 등록부를 변경할수 있다.

```
kcc@linux:~
# mcd
A:/
```

현재 플로피디스크등록부인 a:로 설정되여있다.

mtolls명령문중의 하나인 mdir명령문을 아무러한 지정이 없이 사용하면 현재 mtools에서 지정되여있는 등록부의 내용을 보여준다.

```
kcc@linux:~
# mdir
Volume in drive A is EBOOK
Volume Serial Number is 0341-CC70
Directory for A:/

test                        1093 07-06-2004 19:34 test
dosdir    <DIR>             1093 07-06-2004 19:34 test
                2 file                      1 093  bytes
                   1  454  592 bytes free
```

mcd로 DOS에서와 같이 작업등록부를 변경시킬수 있다.

```
kcc@linux:~

# mcd a:/dosdir

# mcd
A:/dosdir

# mdir
   Volume in drive A is EBOOK
  Volume Serial Number is 0341-CC70
Directory for A:/dosdir

 .    <DIR>     1093 07-06-2004 19:34 test
 ..   <DIR>     1093 07-06-2004 19:34 test
                1 file                     1 093 bytes
                       1 454 592 bytes free
```

참 고

mtools

mtools는 Linux나 UNIX에서 사용하는 MS-DOS명령문이다. MS-DOS에서 사용하는 명령문에 《m》을 앞붙이로 준 형태로 만들어져 있다. 사용법이 MS-DOS와 거의 비슷하므로 DOS를 사용해본 사용자라면 쉽게 사용할수 있다. 뒤에 사용되는 설정값도 거의나 같은것들을 사용하고있다. 파일등록부구분에 있어서만은 《/》를 사용해야 한다.

/etc/mtools.conf파일에서 mtools의 명령문에 사용될 설정을 변경할수 있다. Mtools.conf에서는 BeOS, SCO, Unix, Solarix, SunOS 등 다양한 조작체계들에서의 DOS파일등록부설정방법을 실례로 주고있다. 실례를 참고하면 쉽게 DOS등록부를 설정하여 사용할수 있다.

mcopy

/usr/bin/mcopy

Linux-DOS간의 파일복사

MS-DOS파일과 Linux 및 UNIX파일간의 교환명령문이다.

형식 : mcopy [-설정값][원본파일][복사파일 or 등록부]

-t : 본문파일복사로서 DOS본문형식의 Line/Enter표시가 UNIX형식의 행표시로 변경된다. 또한 UNIX에서 DOS로 복사할 때에는 그 반대로 된다.
-a : 본문(ASCII)파일복사로서 UNIX본문파일형식을 DOS형식으로 변경하거나 그 반대로 변경한다.
-T : 본문(ASCII)파일복사로서 문자모임도 변경해준다.
-b : 일괄(batch)복사
-s : 반복복사
-p : 원본파일의 속성을 가진다.
-Q : 여러 파일복사중 특정파일의 복사가 실패한 경우에 복사를 중지한다.
-n : Linux나 UNIX파일을 덧쓰기할 때 경고통보문이 출력되지 않는다.
-m : 원본의 파일날자를 유지한다.
-v : 복사과정을 자세히 보여준다.

mcopy는 DOS파일과 Linux 및 UNIX 파일사이의 복사명령문이다. DOS파일을 UNIX 및 Linux파일로 복사할수 있다. 물론 Linux 및 UNIX에 가지고있는 cp명령문으로도 복사할수 있지만 mcopy는 DOS와 Linux 및 UNIX에서의 Line/Enter형식이 다른것을 복사중에 변환하는 기능을 가지고 있으므로 편리하게 리용할수 있는 우점이 있다.

플로피디스크의 DosText.txt 파일을 하드웨어에 Linux 및 UNIX의 본문형식으로 보관해보자.

```
# mcopy -t A:DosText.txt /home/ebook
```

그저 복사한 경우에 서로 Line/Enter형식이 다르기때문에 잘못 인식하여 문장끌에 ^M등의 기호가 찍히우게 된다. 우와 같이 mcopy -t명령으로 DOS본문파일의 형식을 Linux 및 UNIX의 본문파일형식으로 변환할수 있다.

mdel

/usr/bin/mdel

MS-DOS파일삭제

MS-DOS파일체계의 DOS파일을 지운다.

형식 : mdel[설정값]DOS 파일

-v : mtools 의 판본을 보여준다.

MS-DOS파일체계에 있는 파일을 삭제하는데 쓰인다.
MS-DOS형식으로 되여있는 플로피디스크에서 test.txt파일을 지운다.

```
kcc@linux:~
# mdir
 Volume in drive A is EBOOK
 Volume Serial Number is 0341-CC70
Directory for A:/

test.txt                   1093 07-06-2004 19:34 test.txt
dosdir        <DIR>        1093 07-06-2004 19:34 test
           2 file                            1 093 bytes
                1   454   592 bytes free
```

test.txt파일을 지운다. 파일등록부는 삭제할수 없으므로 파일등록부를 정리하려고 할 때에는 mtools명령문가운데서 mdeltree 또는 mrd를 사용한다.

```
# mdel a:/test.txt
```

mdir명령으로 삭제된 파일을 확인할수 있다.

```
kcc@linux:~

# mdir
 Volume in drive A is EBOOK
 Volume Serial Number is 0341-CC70
Directory for A:/

dosdir          <DIR>       1093 07-06-2002 19:34 test
                1 file           1,093 bytes
                1 454 592 bytes free
```

참 고

경고음에 의해 핵심부번역의 완료를 알리기

핵심부번역과 같이 작업완료시간이 짧지 않은 경우가 있다.

이런 경우에 다음과 같이 조종탁(Console)의 경고음(Beep)을 사용하여 작업완료를 알리게 할수 있다.

```
#make -s dep; make -s clean, make -s bzimage;\
make -s modules; make -s modules_install; echo -e "/a"
```

소리가 잘 안들리는 환경이라면 X Window를 리용하여 화면에 보여줄수도 있는데 이때에는 다음과 같이 사용한다.

```
# make -s dep; make -s clean; make -s bzimage;\
make -s modules; make -s modules_install; xmessage "complet!"
```

gnome환경인 경우에는 gmessage를 사용할수도 있다.

mdeltree /usr/bin/mdeltree

MS-DOS파일삭제

파일이나 등록부를 삭제한다.

형식 : mdeltree [-v]DOS 파일 등록부

-v: 판본을 보여준다.

mdeltree는 mtools명령문으로서 MS-DOS파일체계에 있는 파일등록부와 아래준위등록부의 모든 파일을 삭제하는데 쓰인다.

mdir명령문으로 MS-DOS파일등록부로 정의된 령역의 파일목록을 확인한다.

```
kcc@linux:~

# mdir
Volume in drive A is EBOOK
Volume Serial Number is 0341-CC70
Directory for A:/

dosdir          <DIR>          1093 07-06-2004 19:34 test
                1 file            1 093  bytes
                1,454,592 bytes free area
```

a:/dosdir등록부의 전체를 한번에 지운다. mdeltree는 파일 혹은 파일등록부를 지울수 있으며 아래준위파일등록부와 파일이 있는 경우에도 지워진다.

```
kcc@linux:~

# mdeltree a:/dosdir

# mdir
 Volume in drive A is EBOOK
 Volme Serial Number is 0341-CC70
Directory for A:/
```

```
No files
                1  455 104 bytes free
```

mdel명령은 파일만을 삭제할수 있는데 비해 mdeltree명령은 등록부와 파일, 아래준위등록부까지 삭제할수 있다. 이와 반면에 mrd명령은 등록부만을 삭제할수 있으며 아래준위등록부나 파일의 삭제는 할수 없다.

참 고

mtools삭제명령문들의 비교

명령문	삭제대상	설　　명
mdel	파일	파일만을 삭제할수 있다.
mdeltree	파일, 등록부	파일과 등록부의 삭제를 다 할수 있다. 아래준위등록부나 파일이 있는 경우에도 모두 삭제할수 있다.
mrd	등록부	등록부만을 삭제할수 있다. 아래준위등록부나 파일이 있는 경우에는 삭제하지 못한다.

mdir

/usr/bin/mdir

MS-DOS파일목록보기

MS-DOS등록부의 파일목록을 보여준다.

형식 : mdir[설정값]DOS 파일(등록부)

-/ : 반복적으로 파일목록을 보여준다.
-w: 파일이름만을 보여준다.
-a : 숨긴 파일을 포함한 모든 파일을 보여준다.
-f : 디스크의 남은 용량을 검사하지 않고 파일목록을 보여준다. 디스크용량을 보여주지 않으므로 목록을 보여주는 속도가 빠르다.
-b : 파일이름과 등록부이름을 한행에 하나씩 간결하게 보여준다.
-v : 판본을 보여준다.

mdir는 mtools명령문으로서 MS-DOS파일체계에 있는 파일목록을 보여준다. 사용법과 설정값은 MS-DOS와 거의나 같다. 경로가 파일등록부가 아닌 경우에는 오유가 발생한다.

```
kcc@linux:~

# mdir a:/
Volume in drive A is EBOOK
Volume Serial Number is 0341-CC70
Directory for A:/

test                     1093 07-06-2004 19:34 test
addtest                  24 07-06-2004 19:34 addtest
addtest2                 15 07-06-2004 19:34 addtest2
dosdir   <DIR>              07-06-2004 19:34 dosdir
               1 file                          1 093 bytes
                     1  454 592 bytes free
```

-a설정값을 리용하면 숨긴파일도 볼수 있다. -ab설정값을 리용하여 숨긴파일을 포함한 모든 파일을 한행씩 화면에 출력해보자.

```
kcc@linux:~
# mdir -ab
A:/test
A:/testhidden
A:/addtest
A:/addtest2
A:/dosdir/
```

mformat

/usr/bin/mformat

MS-DOS파일체계만들기

플로피디스크를 DOS파일형식으로 초기화한다.

형식 : mformat [-f 크기][-1/4/8]구동기
mformat[-설정값]구동기

-f : 초기화되는 DOS파일체계의 크기를 정한다.
-1 : 외면(single-sided)방식으로 초기화한다.
-4 : 량면(double-sided)방식에 360KB로 초기화한다. 《-1》과 같이 쓰이는 경우에 한쪽면이 180KB로 초기화된다.
-8 : 자리길당 8분구(8sector/track)로 초기화한다.
-t 실린더 : 실린더수를 정한다.
-h 헤드 : 헤드수를 정한다.
-n 분구 : 자리길당 분구(sector/track)수를 정한다.
-v 기록권준위 : 기록권준위(volume level)를 써준다.

mformat는 mtools명령문이다. mformat는 UNIX파일체계로 저준위(low level)초기화된 디스크에 MS-DOS파일체계를 추가한다.

초기화하는 플로피디스크의 용량과 특성

용 량	기록면방식	자리길당 분구수	실린더	디스크종류
160k	Single-sided	8 sectors per track	40	5.25
160K	Single-sided	9 sectors per track	40	5.25 DD
320K	double-sided	8 sectors per track	40	5.25 DD
360K	Double-sided	9 sectors per track	40	5.25 DD
720K	Double-sided	9 sectors per track	80	3.5 DD
1200K	Double-sided1	5 sectors per track	80	5.25 HD
1440K	Double-sided1	8 sectors per track	80	3.5 HD
2880K	Double-seded	36 sectors per track	80	3.5 ED

플로피디스크를 1440KB용량의 DOS파일체계로 초기화하고 《ebook》라는 label명을 부여해보자.

```
# mformat -v ebook -f 1440 a:
# mdir a:
 Volume in drive A is EBOOK
 Volume Serinal Number is 08E4-F131
Directory for A:/

No files
                    1  455  104 bytes free
```

초기화된 디스크에 관한 자세한 정보를 보기 위해서는 minfo명령을 리용한다.

【관련명령문】

mtools를 참고할것

minfo

/usr/bin/minfo

MS-DOS파일체계에서 매개변수보기

MS-DOS파일체계의 디스크정보를 보여준다.

형식 : minfo[설정값]구동기

-v : 기동분구(boot sector)의 hexdump 를 보여준다.

minfo는 mtools명령문으로서 MS-DOS파일체계를 사용하는 디스크의 정보를 보여준다. 장치정보와 기동분구정보를 별도로 보여준다.

```
kcc@linux:~

# minfo a:
device information:
====================
   filename= "/dev/fd0"
   sectors per track : 18
heads : 2
cylinders : 80
mformat command line : mformat -t 80 -h 2 -s 18 a:
bootsector information
========================
   banner:"MTOOL398"
sector size : 512 bytes
cluster size : 1 sectors
 ------------중간 생략-----------------------------
 serial number : 08E4F131
    disk label= "PIRANA"
    disk type="FAT12"
```

【관련명령문】

mtools를 참고할것

mkdir

/bin/mkdir

등록부만들기

등록부를 만든다.

형식 : mkdir[설정값]등록부이름

-m, —mode 권한 : 새로 만들어지는 등록부의 권한을 설정한다.
-p, —parents : 상위경로를 함께 만든다.
—help : 도움말을 보여준다.
—version : 판본을 보여준다.

Mkdir는 등록부를 만들 때에 사용하는 명령문이다.

등록부를 만들면서 권한을 부여해줄수 있다. 따로 설정해주지 않는 경우에 기본적으로 755라는 실행권한을 가진다(chmod를 참고할것).

파일의 사용자, 집단, 다른 사용자 등에게 꼭같이 모든 사용권한을 주는 등록부를 만들어 보자.

```
kcc@linux:~

# mkdir -m 777 testdir
# ls -al
-----중간 생략--------
drwxrwxrwx  2 root  root   4096 7월 14 10:47 tesidir
```

-p설정값을 사용하여 웃준위등록부까지 한번에 생성할수 있다.

```
kcc@linux:~

# mkdir -p subdir1/subdir2
# ls -al subdir1
subdir1:
합계 12
drwxr-xr-x  2  root  root  4096 7월 11 07:39.
drwxr-x—-  2  root  root  4096 7월 11 07:39 ..
drwxr-xr-x  2  root  root  4096 7월 11 07:39 subdir2
```

mke2fs

/sbin/mke2fs

ext2파일체계를 생성하는 명령문

Linux파일체계를 생성한다.

형식 : mke2fs[설정값]장치명

-b bytes : 블로크의 크기를 지정한다.
-c : 파일체계를 생성할 때에 불량블로크를 검사한다.
-f bytes : fragement의 크기를 지정한다.
-i bytes : inode당 바이트수(bytes-per-inode)를 정한다. 기본값은 4096bytes이며 최소값은 1024bytes이다.
-l 파일명 : 파일에서 불량블로크를 검사한다.
-m % : super user에게 예약해둔 블로크의 백분률(%)을 정한다. 기본값은 5%이다.
-q : 출력없이 실행한다. 스크립트안에서 사용되여 출력을 내보내지 않는다.
-v : 파일체계의 생성과정을 자세히 보여준다.
-F : 지정한 장치가 블로크장치가 아니라도 mke2fs를 무조건 실행하도록 한다.
-S : 관리정보블로크(super block), 집단서술자(descriptor)만을 사용한다.

mke2fs명령은 디스크에 새로운 Linux파일체계를 만들때에 쓰인다. 이 명령문은 mkfs -t ext2명령과 같다.

```
kcc@linux:~

# mke2fs /dev/hda4
mke2fs 1.23, 15-Aug-2003 for EXT@FS 0.5b, 95/08/09
Fiesystem label=
OS type: Linux
Block size=4096 (log=2)
------중간 생략-----------------
Writing inode tables : done
Writing Superblocks and filesystem accounting information : done
This filesystem will be automatically checked every 31 mount or
180 days, whichever comes first. Use tune2fs -c or -to override.
```

새로 생성된 파일체계는 mount하여 사용할수 있다. /etc/fstab파일에 등록하면 체계를 다시 시작할 때마다 새로 mount하는 수고를 들이지 않고도 사용가능하다.

【관련명령문】

mkfs : Linux파일체계를 생성하는 명령문

참 고

편집(journaling)파일체계

이전에 대부분의 Linux체계에서는 파일체계로서 ext2를 사용하였다. 아는바와 같이 체계가 비정상적으로 완료한 경우, 기동하였을 때의 파일체계를 검사조회한다. 이것은 파일작성하여 디스크에 기록된 다음 디스크의 색인(index)정보를 갱신하기 전에 체계가 비정상적으로 꺼졌을 때, 이 파일에 대한 정보를 알수 없기때문인데 디스크의 용량이 커져감에 따라 이러한 작업은 상당히 오랜 시간이 걸리게 되였다.

편집파일체계는 파일을 작성하고 디스크색인정보가 갱신되지 않는 경우에도 자체적인 리력(log)과 디스크색인을 비교하여 오유를 수정하므로 안전하면서도 대용량의 디스크를 사용할 때에 특히 편리하다. 현재 Linux에서 사용할수 있는 편집파일체계로서는 Reiserfs, JFS(IBM), XFS(SGI), ext3 등이 있다.

fstab파일

fstab파일은 체계가 기동할 때에 자동적으로 mount하는 파일체계설정을 가지고 있다. 그러므로 기동할 때마다 반복적으로 디스크를 추가하지 않고 자동으로 mount하여 사용하기 위해서는 /etc/fstab파일을 수정하여야 한다. /etc/fstab파일의 열기는 편집기를 리용하여 한다. mke2fs와 같은 명령문에 의해 파일체계가 새로 만들어진 하드디스크를 체계가 기동할 때마다 자동적으로 읽혀져 사용하도록 하려면 아래와 같이 새로 추가할 파일체계의 정보를 적어준다. 이때 mount하여 사용할 등록부인 /home/Backup은 만들어져 있어야 한다.

```
kcc@linux:~

LABEL=1      /              ext3     defaults          1 1
None         /dev/pts       devpts   gid=5,mode=620    0 0
none         /proc          proc     defaults          0 0
/dev/hda3    /home          ext2     defaults          1 2
/dev/hda2    swap           swap     defaults          0 0
/dev/hda4    home/Backup    ext2     defaults          0 0
```

보관한 fstab의 정보는 체계가 기동할 때 자동적으로 읽어들여 mount하며 체계를 완료하지 않고 사용하려면 mount -a명령을 사용한다. df -T명령으로 현재 체계에 mount되여있는 파일체계식별이름을 확인한다.

```
kcc@linux:~

#>mount -a
#>df -T
Filesystem Type 1K-blocks Used    Available  Use%   Mounted on
/dev/hda1  ext3  5036284  3072580 1707872   65%     /
/devhda3   ext2  7961676  3600660 3956580   48%     /home
/devhda4   ext2  2696024    32828  2526244   2%     /home/Backup
```

mkfs

/sbin/mkfs, /sbin/mkfs.msdos, /sbin/mkfs.reiserfs, /sbin/mkfs .bfs, /sbin/mkfs.ext2/sbin/mkfs.vfat

Linux파일체계생성명령문

파일체계를 만드는 명령문이다.

형식 : mkfs[설정값]장치명[크기]

-V : 정보를 자세히 보여준다.
-t 파일체계형식 : 만들어지게 되는 파일체계의 종류를 지정한다.
-c : 파일체계를 생성하기전에 먼저 불량블로크를 검사한다.
-l 파일이름 : 지정한 파일명으로부터 불량블로크목록을 읽는다.
-v : 작업상태와 결과를 자세히 보여준다.

mkfs는 파일체계를 만들어주는 명령문으로서 ext2, ext3, vfat등의 다양한 파일체계를 만들수 있게 한다.

Primary Slave구획의 구성을 보자. fdisk -l명령으로 확인할수 있다.

```
kcc@linux:~

# fdisk -l /dev/hdb
Disk /dev/hdb: 255 heads, 63 sectors, 2491 cylinders
Unrts=cylinders of 16065 * 512 bytes

  Device Boot    Start    End     Blocks     Id    System
/dev/hdb1            1    1275    10241406   83    Linux
/dev/hdb2         1276    2491     9767520   83    Linux
```

dev/hdb는 Linux System으로 2개의 구획이 나누어져 있다.

/dev/hdb2구획을 ext2파일체계로 생성하여 보자.

새로 파일체계를 만들 device명을 입력한다.

```
kcc@linux:~

# mkfs -t ext2 /dev/hdb2
mke2fS 1.23 15-Aug-2003 for EXT2 FS 0.5b,  95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
1221600 inodes, 2441880 blooks
122094 blocks (5.00%) reserved for the super user
First data block=0
75 block groups
32768 blocks per group, 32768 fragments per group
16288 inodos per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Writing superblocks and filesystem accounting informatfon: done

This filesystem will be automatically checked every 25 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i override.
```

【관련명령문】

mke2fs : ext2파일체계생성명령문

mkdosfs : msdos파일체계생성명령문

mkswap : Linux swap령역을 지정하는 명령문

mknod

/bin/mknod

특수파일만들기

특수파일을 만든다.

형식 : mknod[설정값]파일이름{bcu}주번호 종속번호
mknod[설정값]파일이름 p

-m, —mode 권한 : mode는 사용자권한을 의미한다. chmod명령문을 참고하여 권한을 사용한다.
—help : 도움말을 보여준다.
—version : 판본을 보여준다.

/dev/sda 파일을 보자.

```
kcc@linux:~

# ls -al /dev/sda*
brw-rw----  1 root  root   8,   0 1월 23 14:55  /dev/sda
brw-rw----  1 root  root   8,   1 1월 23 14:55  /dev/sda1
brw-rw----  1 root  root   8, 10 1월 23 14:55  /dev/sda10
brw-rw----  1 root  root   8, 11 1월 23 14:55  /dev/sda11
brw-rw----  1 root  root   8, 12 1월 23 14:55  /dev/sda12
brw-rw----  1 root  root   8, 13 1월 23 14:55  /dev/sda13
brw-rw----  1 root  root   8, 14 1월 23 14:55  /dev/sda14
brw-rw----  1 root  root   8, 15 1월 23 14:55  /dev/sda15
brw-rw----  1 root  root   8,   2 1월 23 14:55  /dev/sda2
brw-rw----  1 root  root   8.   3 1월 23 14:55  /dev/sda3
brw-rw----  1 root  root   8,   4 1월 23 14:55  /dev/sda4
brw-rw----  1 root  root   8,   5 1월 23 14:55  /dev/sda5
brw-rw----  1 root  root   8,   6 1월 23 14:55  /dev/sda6
brw-rw----  1 root  root   8,   7 1월 23 14:55  /dev/sda7
brw-rw----  1 root  root   8,   8 1월 23 14:55  /dev/sda8
```

[주의]
/dev/had는 첫번째 IDE하드디스크, /dev/sda는 첫번째 SCSI하드디스크이다.

brw-rw----에서 b는 블로크형장치(block device)를 c는 문자형장치(character device)를 나타낸다.

여기서 블로크형장치는 디스크, CD-ROM 등과 같이 매체를 리용한 블로크단위검색을 하며 문자형장치는 통신포구, 인쇄기포구 등과 같이 1byte씩 순차적접근을 하는 장치를 말한다.

8,0은 주번호(major number)와 종속번호(minor number)를 나타낸다.

SCSI디스크의 주번호는 8로 시작하며 IDE디스크는 3으로 시작한다.

sda1, sda2의 주번호는 8로서 같으며 종속번호는 1, 2로 증가한다. dev/아래에 있는 하드웨어장치파일은 mknod명령이나 /dev/MAKEDEV스크립트를 리용한다.

mkswap

/sbin/mkswap

swap령역지정

Linux swap령역을 지정하는 명령문이다.

형식 : mkswap[설정값]장치이름[블로크크기]

-c : swap령역을 만들기전에 먼저 불량블로크를 검사한다.

mkswap명령은 지정한 특정장치나 파일을 swap령역으로 지정한다.

아래에 512MB크기의 Linux swap파일(교환파일이라고도 한다)을 만드는 실례를 주었다.

512MB의 swap파일만들기

```
kcc@linux:~
# dd if=/dev/zero of=/swap bs=1024 count=524288
# mkswap /swap 524288
# sync
# swapon /swap
```

free명령으로 작성된 swap파일을 확인한다.

```
kcc@linux:~
# free
```

swap를 해제하려면 swapoff명령을 사용한다.

```
kcc@linux:~
# swapoff /swap
# free
```

mlabel

/usr/bin/mlable

MS-DOS디스크식별표식만들기

새로운 기록권식별표식(volum label)을 만들거나 기록권계렬번호(volume serial number)를 만든다.

형식 : mlabel[설정값]구동기:

-c : 현재의 label을 지운다.
-s : 현재의 label을 보여준다.
-n : 새 계렬번호를 란수로 만들어 낸다.
-N 계렬 : 새로운 계렬번호를 지정하여 만든다. 16진수의 8자리로 만든다.

mlabel은 mtools명령문으로서 이것은 MS-DOS에서의 label명령문과 같은 기능으로 쓰인다.
a 구동기의 이름을 지정한다.

```
kcc@linux:~

# mlabel a:
Volume label is EBOOK
Enter the new volume label : first

# mlabel -s a:
Volume label is FIRST
```

기록권계렬번호의 변경은 16진수의 8자리로 이루어진다.

```
# mlabel -N 01234567 a:
```

변경된 내용은 mdir를 리용하여 확인할수 있다.

```
kcc@linux:~

# mdir
Volume in drive A has FIRST
Volume Serial Numher is 0123-4567
Directory for A:/

No files
        1 457 664 bytes free
```

【관련명령문】

mtools 명령문 참고

참 고

DOS (Disk Operating System)

DOS는 현재 거의나 쓰이지 않는 PC조작체계이다. 최초의 DOS는 PC-DOS라고 불리웠는데 Microsoft회사가 개발하여 IBM에 출하한것이였다. 당시 마이크로쏘프트회사는 MS-DOS라고 이름지은 제품을 직접 시장에 팔수 있는 권한을 가졌다. PC-DOS와 MS-DOS는 내용면에서 거의나 같은것이기때문에 대부분의 사용자들은 이 두가지를 모두 《DOS》라고 불렀다. 그밖에 DR-DOS도 사용되였으나 Windows조작체계가 나타난후 인차 없어졌다.

DOS는 도형처리방식을 지원하지 않고 명령문입력방식에 기초한것으로서 사용하기에도 불편한 점이 많았다. Windows조작체계의 초기판본은 실제로는 MS-DOS에서 돌아가는 일종의 편의프로그람과 같은 형태로부터 시작되였으며 Windows조작체계에서는 지금도 DOS화면을 볼수 있다.

mmd

/usr/bin/mmd

MS-DOS파일체계등록부의 만들기

MS-DOS등록부를 만든다.

형식 : mmd[설정값]등록부명

mmd는 mtools 명령문으로서 MS-DOS 파일등록부를 생성하는데 쓰인다.

```
kcc@linux:~

# mmd a:/testdir

# mdir a:/
 Volume in drive A is EBOOK
 Volume Serial Number is 0341-CC70
Directory for A:/

Testdir   <DIR>  1093 07-06-2004 19:34 testdir
                 1 file                       1 093 bytes
                       1 454 592 bytes free
```

【관련명령문】

mtools명령문 참고할것

mmount

/usr/bin/mmount

MS-DOS파일체계적재

MS-DOS디스크를 적재(mount)한다.

형식 : mmount DOS 구동기 mount 위치

mmount는 mtools명령문으로서 MS-DOS파일등록부를 마운트(mount)한다.

mtools가 사용하고있는 플로피디스크구동기인 A:구동기를 /mnt/floppy파일등록부에 마운트(mount)하여 사용한다.

```
# mmount a: /mnt/floppy
```

mmount로 MS-DOS구동기를 등록하기 위해서는/etc/fstab 또는 /etc/mtab파일에 등록되여있어야 한다.

만약 등록되여있지 않으면 mount위치의 등록부를 파일내용에서 찾을수 없다는 오유통보문을 출력한다. 이때에는 아래와 같은 행을 /etc/fstab파일에 추가한다.

```
/dev/fd0   /mnt/floppy            vfat   detaults   0 0
```

【관련명령문】

mtools를 참고할것

mmove

/usr/bin/mmove

MS-DOS파일이동/이름변경

MS-DOS의 파일이나 등록부를 이동하거나 이름을 변경한다.

형식 : mmove [설정값] 파일 [이동위치/파일이름]

-V : 판본을 보여준다.

mmove는 mtools명령문으로서 파일이나 파일등록부를 이동하거나 이름을 변경시킨다. mmove명령은 mtools를 통해 지정된 구동기에로의 이동만을 할 수 있다. 구동기의 지정이 없는 경우에는 mtools가 지정하는 파일등록부를 가리킨다.

파일등록부지정은 mcd명령문을 리용하여 변경한다.

```
kcc@linux:~

# mcd a:/directory

# mmove a:/dirtest testdir

# mdir
Volume in drive A is EBOOK
Volume Serial Number is 0341-CC70
Dirctory for A:/directory

.          <DIR>      07-05-2004 15:21
..         <DIR>      07-05-2004 15:21
testdir    <DIR>      07-06-2004 19:34 testdir
                3 file          3 093 bytes
                      1 454  592 bytes free
```

【관련명령문】

mtools를 참고할것.

modprobe

/sbin/modprobe

핵심부모듈적재

핵심부(kenel)에 모듈을 적재한다.

형식 : modprobe [설정값]파일 [이동위치/파일이름]

모듈적재
modprobe [-a-n -v] [-C config] [-t 형태] 패턴 OR 모듈명1 모듈명2...
모듈목록
modprobe [-l] [-C config][-t 형태] 패턴
설정보기
modprobe [-C config] -c
모듈제거 혹은 자동제거
modprobe [-C config]-r[모듈명…]
-a, —all : 모듈과 의존성으로 관련된 다른 모듈도 함께 적재한다.
-c, —showconfig : 현재의 설정을 보여준다.
-d, —debug : 오유수정정보를 보여준다.
-h, —help : 사용법을 보여준다.
-k, —autoclean: autoclean 모듈을 적재한다.
-l, —list : 핵심부모듈들을 보여준다.
-n, —show : Don't actually perform the action
-q, —quiet : Quiet operation
-r, —remove : 모듈 또는 autoclean 모듈을 제거한다.
-s, —syslog L : 통보문을 syslog에로 보낸다.
-t, —type moduletype : Only look for modules of this type
-V, version : 판본정보를 보여준다.
-C, —config configfile : /etc/modules.conf 파일대신에 지정한 설정 파일을 리용한다.

modprobe명령은 insmod보다 높은 수준의 조종을 할수 있는 모듈조종 명령문이다.

RedHat배포판인 경우에는 모듈의 파일등록부는 /lib/modules/핵심부판본/kernel/에 있다.

```
kcc@linux:~
# cd /lib/modules/2.4.13-1hl//kernel/
# ls
arch  drivers  fs  net
```

현재 주기억기에 적재된 모듈들을 먼저 보기로 하자.

```
kcc@linux:~
# lsmod
Module          size     Used by   Tainted:
P
ip_tables      11200      0 (unused)
ymfpci         42660      0 (autoclean)
uart401        6560       0 (autoclean)
[ymfpci]
sound            59052    0 (autoclean)
[uart401]
ac97_codec     9504       0 (autoclean)
[ymfpci]
soundcore       4324      4 (autoclean)
[ymfpci sound]
8139too         17440     2
ext3             61600    2 (autoclean)
jbd              40452    2 (autoclean)[ext3]
usb-uhci        21764     0 (unused)
usbcore         51744     1 [usb-uhci]
```

ip_tables모듈, 음향관련모듈, LAN Card모듈, ext3모듈, USB관련모듈등이 있다. 그중 ip_tables모듈을 rmmod명령을 사용하여 주기억기에서 제거해보자.

```
# rmmod ip_tables
```

insmod명령으로 다시 ip_tables모듈을 적재한다.

```
# insmod /lib/modules/2.4.13-1h/kernel/net/ipv4/netfilter/ip_tables.o
```

depmod명령으로 매 모듈에 대한 의존성을 검사하여 모듈자료기지를 생성한다. 생성된 파일은 lib/modules/핵심부판본/modules.dep으로 생성된다.

```
kcc@linux:~
# ls/lib/modules/핵심부판본
build          modules.generic_string  modules.parportmap  modules.usbmap
kernel         modules.ieee1394map     modules.pcimap      pcmcia
modules.dep    modules.isapnpmap       modules.pnpbiosmap ㄴ
```

이 modules.dep모듈자료기지로 modprobe명령을 사용할수 있다.

례를 들어 아래의 ipt_MASQUERADE.o모듈은 먼저 ip_conntrack.o, ip_tables.o, ip_table_nat.o 모듈을 적재해야 하지만 modprobe명령은 자동적으로 의존성을 검사하여 동시에 모듈을 적재하여준다.

아래의것은 modules.det 파일의 내용중 ipt_MASQUERADE.o와 관련된 내용이다.

```
/lib/modules/2.4.13-1hl/kernel/net/ipv4/netfilter/ipt_MASQUERADE.o:
/lib/modules/2.4.13-1hl//kernel/net/ipv4/netfilter/ip_conntrack.o\
/lib/modules/2.4.13-1hl/kernel/net/ipv4/netfilter/ip_tables.o
/lib/modules/2.4.13-1hl/kernel/net/ipv4/netfilter/iptable_nat.o
```

그러면 ipt_MASQUERADE모듈을 modprbe를 사용하여 적재해보자.

여기에서는 의존성이 있는 ipt_conntrack.o ip_tables.o, iptable_net.o 모듈을 함께 적재하는것도 확인할수 있다.

```
kcc@linux:~

# modprobe ipt_MASQUERADE
# lsmod
Module                Size  Used by Tainted: P
ipt_MASQUERADE              1728   0 (unused)
iptable_nat                 17364 0 [lipt_MASQUERADE]
ip_conntrack                16364 1 [lipt_MASQUERADE iptable_nat]
ip_tables                   11200 4 [ipt MASQUERADE iptable_nat]
ymfpci                      42660 0 (autoclean)
uart401                     6560  0 (autoclean) [ymfpci]
sound                       59052 0 (autoclean) [uart401]
ac97_codec                  9504  0 (autoclean) [ymfpci]
soundcore                   4324  4 (autoclean) [ymfpci sound]
8139too                     17440 2
ext3                        61600  2 (autoclean)
jbd                         40452  2 (autoclean) [ext3]
usb-uhci                    21764  0 (unused)
usbcore                     51744  1 [usb-uhci]
```

more /bin/more

화면단위로 파일보기

파일을 요구하는 형식으로 보여준다.

형식 : more[설정값][-행][+/표현식][+행][파일이름]

-d : more지령은 선택건을 입력받아야 할 때에 《[Press space to continue, <q> to quit]》 내용을 보여주며 글자가 잘못 입력되였을 때에는 《[Press <n> for instructions]》라는 본문내용을 보여준다.
-p : 화면을 모두 지우고 본문을 보여준다.
-s : 여러개의 빈 행을 한행으로 보여준다.
-u : 밑줄속성을 가진 문자를 그대로 보여준다.
+/표현식 : 정규표현식이 발견되는 부분부터 보여준다.
- 행 : 한 화면에 출력할 행수를 지정한다.
+행 : 지정한 행부터 보여준다.

more명령은 한페지이상을 초과하는 내용을 입력받아 페지별로 볼수 있게 한다. 또한 검색과 요구하는 페지에로의 이동도 할수 있다.

more상태에서 사용할수 있는 명령문

H	more의 도움말을 출력
Z	다음 페지를 출력
Enter Key	한행을 출력
d, ^D	반페지를 출력
q, Q	완료
F	다음 페지에로
b, ^B	이전 페지에로
/검색할 단어	단어검색

한 페지가 넘는 ps명령을 입력받아 파이프(|)를 리용하여 more로 보여주는 실례를 아래에 주었다.

```
kcc@linux:~

# ps aux | more
USER      PID %CPU %MEM VSZ RSS TTY STAT START TIME
COMMAND
root        1 0.0  0.0  1384   68 ?  S    Jun02 0:04 init
root        2 0.0  0.0     0    0 ?  SW   Jun02 0:00 [keventd]
root        3 0.0  0.0     0    0 ?  SW   Jun02 0:00[kapm-idled]
root          4    0.0  0.0     0     0 ?  SWN Jun02 0:00
[ksoftirqd_CPU0]
root       5 0.0 0.0      0     0 ?    SW Jun02 0:05 [kswapd]
root       6 0.0 0.0      0     0 ?    SW Jun02 0:00 [kreclaimd]
root       7 0.0 0.0      0     0 ?    SW Jun02 0:00 [bdflush]
root       8 0.0 0.0      0    0 ?     SW Jun02 0:00 [kupdated]
root       9 0.0 0.0     0    0 ?     SW Jun02 0:00 [mdrecoveryd]
root      63 0.0 0.0     0    0 ?     SW Jun02 0:00 [khub]
root     163 0.0 0.0     0    0 ?     SW Jun02 0:00 [kjournald]
root     164 0.0 0.0     0    0 ?     SW Jun02 0:00 [kjournald]
root     703 0.0 0.0 1444  160 ?     S Jun02 0:00 syslogd -m 0
root     708 0.0 0.1 1988  172 ?     S Jun02 0:00 klogd -2
root     774 0.0 0.0 2644   20 ?     S Jun02 0:01 sshd
root     810 0.0 0.0 1968    0 ?     SW Jun02 0:00 bin/sh /usr/bin/
mysql 854 0.0 0.0 28208  24 ?    S Jun02 0:00 /usr/libexec/mysq0
mysql 856 0.0 0.0 28208  28 ?    S Jun02 0:00 /usr/libexec/mysq
mysql 857 0.0 0.0 28208  28 ?    S Jun02 0:00 /usr/libexec/mysq
mysql 858 0.0 0.0 28208  28 ?    S Jun02 0:00 /usr/libexec/mysq
root  859 0.0 0.4  81192  564 ?   S Jun02 0:00 /usr/sbin/httpd-
-More-
```

mount

/bin/mount

장치련결

파일체계를 마운트(mount)하여 사용할수 있게 만든다.

형식 : mount -a[설정값]장치등록부

-h : 도움말을 보여준다.

-V : 판본정보를 보여준다.

-a : fstab에 서술되여있는 모든 파일체계를 mount한다. noauto설정값은 자동mount에서 제외한다.

-v : 자세한 설명을 보여준다.

-f : 실제적인 체계호출은 하지 않고 mount할수 있는가를 검사한다.

-n : /etc/mtab파일에 쓰기작업을 하지 않고 mount한다. /etc가 읽기전용파일체계인 경우에 필요하다.

-r : 읽기만 가능하도록 mount한다. -o ro 설정값과 같다.

-w : 읽기/쓰기방식으로 mount한다. 기본적인 설정값이다. -o rw설정값과 같다.

-t **파일체계** : 파일체계형식을 지정한다.

-o : -o설정값뒤에 두점(:)으로 다음 설정값들가운데서 필요한것을 선택하여 사용한다.

- async : 파일체계에 대한 I/O가 비동기적으로 이루어지도록 한다.
- Auto : -a설정값으로 mount한다.
- defaults : rw, suid, dev, exec, auto, nouser, async를 기본설정값으로 한다.
- dev : 파일체계상의 문자, 블로크특수장치를 해석한다.
- exec : 2진파일의 실행을 허가한다.
- noato : -a설정값으로는 mount되지 않는다.
- nodev : 파일체계상의 문자, 블로크장치에 대한 해석을 하지 않는다.
- noexec : 파일체계에 실행권한을 주지 않는다.
- nosuid : set-UID, set-GID 를 무시하도록 한다.
- nouser : 일반사용자는 mount를 못하도록 한다.
- remount : 이미 mount된 파일체계를 다시 mount한다.

- rw : 읽기 및 쓰기를 할수 있도록 적재(mount)한다.
- suid : set-UID, set-GID가 효력을 나타낼수 있도록 해준다.
- vsync : 파일체계에 대한 I/O가 동기적으로 이루어지게 한다.
- user : 일반 사용자도 적재(mount)할수 있도록 허용한다.

Linux에서는 장치도 포함하여 모든것들을 /아래에 하나의 등록부로서 인식한다. etc/fstab에 설정하여 기동시에 자동적으로 mount하게 한다.

```
kcc@linux:~

# cat /etc/fstab
LABEL=/        /          ext2    defaults         1 1
LABEL=/boot    /boot      ext3    defaults         1 2
none           /dev/pts   devpts  gid=5,mode=620   0 0
LABEL=home     /home      ext3    defaults         1 2
none           /proc      proc    defaults         0 0
none           /dev/shm   tmpfs   defaults         0 0
LABEL=/tmp     / tmp      ext2    defaults         1 2
LABEL=/var     /var       ext2    defaults         1 2
/dev/hda5      swap       swap    defaults         0 0
/dev/hda6      swap       swap    defaults         0 0
/dev/hdb1      hdb1       ext2    defaults         0 0
/dev/hdb2      /hdb2      ext2    defaults         0 0
/dev/hdc1      /hdc1      ext2    defaults         0 0
/dev/cdrom   /mnt/cdrom iso9660 iocharste=cp949,codepage=949,noa
uto, owner,kudzu,ro 0 0
/dev/fd0     /mnt/floppy   auto iocharset=cp949,codepage=949,noauto,
owner,kudzu 0 0
```

우와 같이 매 구획을 지정하여 기동시에 자동적으로 마운트(mount)하도록 하였다. 보통/etc/fstab는 [장치명][mount할 등록부][파일체계][설정값][dump] [기동시에 파일체계검사순서]로 지정하게 된다.

여기에서 dev/hdb1구획은 hdb1등록부로 ext2파일체계를 자동mount 하고 기정설정값으로 dump는 실행하지 않으며 기동할 때에 파일체계를 검사하지 않도록 설정하였다.

아래에 자세한 설명을 주었다.

장치명	마운트될 블로크장치 또는 원격파일체계를 적어준다.
mount할 등록부	파일체계를 마운트할 파일등록부를 지정한다.
파일체계	파일체계의 류형을 지정한다.
설정값	파일체계에 관련된 mount설정값을 지정한다.
Dump	dump명령에 의해 출력표시될 파일체계를 지정한다. 만일 0값이면 dump가 필요없다고 판단한다.
기동시에 파일 체계 검사순서	기동시에 파일체계의 이상여부를 확인한 후, 이상이 있는 경우에는 자동체계검사를 할것인가를 지정한다. 0이면 검사하지 않기, 1이면 /구획을 검사, 그외에 한해서는 2이상의 우선권순위를 지정해주면 된다. 모두를 1로 지정하여도 문제로 되지는 않는다.

-l설정값은 마운트한 파일체계의 정보를 보여준다.

```
kcc@linux:~

# mount -1
/dev/hda8 on / type ext2 (rw) [/]
none on /proc lype proc(rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
/dev/hda1 on /boot type ext3 (rw) [/boot]
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda3 on /home type ext3 (rw) [/home]
none on /dev/shm type tmpfs (rw)
/dev/hda2 on /tmp type ext2 (rw) [/tmp]
/dev/hda7 on  /var type ext2(rw) [/var]
/dev/hdb1 on /hdb1 type ext2 (rw) []
/dev/hdc1 on hdc1 type ext2 (rw) [/hdc]
/dev/cdrom on /mnt/cdrom type iso9660 (ro,nosuid,nodev,iocharset
=cp949.codepage=949)
//61.40.233.203/jkwoo on /mnt/jkwoo type smbfs (0)
//61.40.233.203/marketing on /mnt/marketing type smbfs (0)
/dev/hdb2 on hdb2 type ext2 (rw) []
```

/dev/hdc1 파일체계를 해제(unmount)하여보자.

```
# umount /hdc1
```

다시 /dev/hdc1파일체계를 마운트(mount)하여보자

```
# mount /dev/hdc1 hdc1
```

【관련명령문】

/etc/fstab : 적재할 파일체계와 설정값을 가지고있는 파일

/etc/mtab : 현재 mount된 파일체계와 설정값을 가지고있는 파일

unmount : 설치해제명령문

mrd

/usr/bin/mrd

MS-DOS등록부삭제

MS-DOS등록부를 삭제하는 명령문이다.

형식 : mrd[설정값]MSDOS 등록부

-v : 판본을 보여준다.

mrd는 mtools 명령문으로서 MS-DOS의 등록부삭제에 쓰인다. 등록부내에 파일이 있거나 아래준위등록부가 있는 경우에는 삭제되지 않는다. 또한 파일도 삭제하지 못한다. 이때에는 mdeltree 명령을 사용하면 아래준위등록부나 파일도 모두 삭제할수 있다.

```
kcc@linux:~

# mdir a:/
 Volume in drive A is EBOOK
 Volume Serial Number is 0341-CC70
Directory for A:/
onlydir   <DIR>   07-03-2004 8:22 onlydir
subdir   <DIR>   07-06-2004 19:34 testdir
       2file                2 142 bytes
 1 454 592 bytes free

# mrd a:/*
Dirrectory A:testdir non empty
# mdir a:/
 volume in drive A is EBOOK
 Volume Serial Number is 0341-CC70
Directory for A:/

subdir    <DIR>         07-06-2004 19:34 testdir
          1 file                     182 bytes
          1,454,592 bytes free
```

mren

/usr/bin/mren

MS-DOS파일이름변경

MS-DOS파일이름을 변경한다.

형식 : mren [설정값]파일이름[변경할 파일이름]

-a : 중복되는 이름과 비슷한 파일이름으로 자동변경한다.
-A : 중복되는 이름과 비슷한 파일이름으로 자동변경한다(모든 파일에 적용).
-r : 중복되지 않는 이름으로 변경한다.
-R : 중복되지 않는 이름으로 변경한다. (모든 파일에 적용)
-o : 덧쓰기를 한다.
-O : 덧쓰기를 한다(모든 파일에 적용)
-s : 지령을 적용시키지 않는다.
-S : 지령을 적용시키지 않는다. (모든 파일에 적용)
-V : 판본을 보여준다.

mren은 mtools명령문으로서 MS-DOS파일이나 등록부의 이름을 변경하는데 사용된다.

```
kcc@linux:~

# mdir a:/
 Volume in drive A is EBOOK
 Volume Serlal Number is 0341-CC70
Directory for A:/
Test              1093   07-05-2004 12:31 test
Test1             1093   07-05-2004 12:31 test1
dirtest      <DIR>      07-06-2004 19:34 dirtest
                  3 file              2,186 bytes
                  1,454,592 bvtes free
```

Dirtest 파일등록부이름을 testdir로 변경해보자.

```
# mren a:/dirtest a:/testdir
```

test라는 파일이름을 test1로 변경해보자. 파일이름이 같은 경우, 자동적으로 다른 이름으로 변경되도록 -a설정값으로 지시한다.

```
# mren -A a:test a:test1
```

mdir명령을 사용하면 변경된 파일과 파일등록부이름을 확인할수 있다.

```
kcc@linux:~
# mdir a:/
 Volume in drive A is EBOOK
 Volume Serial Number is 0341-CC70
Directory for A:/
test1-1          1093   07-07-2004 15:31 test
test1            1093   07-07-2004 12:31 test1
direst   <DIR>          07-07-2004 15:31 dirtest
                 3 file                 2,186 bytes
                 1,454,592 bytes free
```

【관련명령문】

mtools를 참고할것.

mtype

/usr/bin/mtype

MS-DOS파일내용보기

MS-DOS파일의 내용을 보여준다.

형식 : mtype [설정값]파일이름

-t : 본문파일을 보여준다. Carriage return/line feed형식을 line feed 형식으로 변경하여 출력한다.
-v : 판본을 보여준다.

mtype는 mtools명령문으로서 MSDOS파일을 보여주는 명령문이다.

```
# mtype -t a:/test.txt
```

파일의 내용이 출력된다. -t는 MS-DOS의 행열기형식을 Linux 및 UNIX의 행열기형식으로 변경하여 출력하도록 한다.

참 고

Carriage return/line feed

한행의 출력을 끝내고 다음 행의 첫 부분으로 보내는 기능이다. carriage return(CR)은 복귀기능으로서 현재 행의 가장 앞으로 가도록 하며 line feed는 바로 다음 행의 같은 위치에로 보내는 기능을 한다. 두 기능이 합쳐져 행의 마지막까지 출력하고 다음 행의 가장 앞부분으로 이동하게 된다.

mv

/bin/mv

파일이동

파일 및 파일등록부를 이동하거나 이름을 변경할 때에 사용한다.

형식 : mv[설정값]source dest
mv[설정값]source directory
mv -d[설정값] target directory=directory sourceenv
[설정값][변수=값…] [명령[인수…]]

-b : 예비보관 파일을 생성한다.
-i —interactive : 기본적으로 .bashrc에 alias되여있는 설정값으로 이동시에 동일한 파일이 있는 경우, 사용자에게 확인을 하도록 한다.
-v —verbose : 이동하기 전에 각 파일이름을 출력한다.
-u —update : 이동할 파일이 새로운 파일인 경우에만 이동한다.

mv명령문은 파일이동과 이름변경을 할수 있다. 원천(source)과 대상(dest)의 이름이 다르면 이름을 변경하는것과 같다. 이동할 파일이 여러개인 경우에는 이동방식으로만 동작한다.

아래에 rename의 기능을 하는 mv명령문의 사용을 실례로 보여주었다.

```
kcc@linux:~
# ls
The Palace of Versailes.mp3 Rare Bird.mp3
# mv The Palace of Versailles.mp3 mvtest.mp3
```

현재 등록부의 mp3파일을 모두 /root등록부아래에로 이동시켜 보자.

```
kcc@linux:~
# mv *.mp3 /root/
# ls root/*.mp3
mvtest.mp3  Rare Bird.mp3
```

N

namei /usr/bin/namei

련결원본파일의 경로보기

지정한 파일의 형태와 상세한 정보를 보여주는 명령문이다.

형식 : namei[설정값]경로이름[경로이름…]

-x : 적재(mount)된 파일체계의 경로에 대해서는 'd' 대신에 'D'를 표시한다.
-m : rwxrx-rx의 형태와 같이 권한을 보여준다.

namei명령은 절대경로로 지정한 파일이나 파일등록부의 이름을 지정한다. 결과로서는 절대경로인 /등록부부터 아래준위등록부까지의 경로등록부를 하나하나 보여주고 지정된 파일의 형태가 기호파일인가, 블로크장치인가, 일반파일인가 등의 정보를 보여준다.

출력되는 행의 첫 부분에 이 파일의 형태를 알리는 문자를 표시하는데 그것들을 아래에 주었다.

- f:= 입력된 경로이름
- d= 경로
- l= 기호련결(련결대상으로 되는 파일은 다음에 들여쓰기로 보여준다.)
- s= 소케트
- b= 블로크장치
- c= 문자장치
- -= 일반파일
- = 알수 없는 파일

httpd.conf파일을 한번 보자. 입력된 경로이름이 f: 다음에 나타나 있고 d에 의해서 아래준위의 /, etc, httpd conf가 등록부이라는것을 알려준다.

그리고 -에 의해 httpd.conf가 일반파일이라는것을 알수 있다.

```
kcc@linux:~
# namei /etc/httpd/conf/httpd.conf
f: /etc/httpd/conf/httpd.conf
d /
d etc
d httpd
d conf
- httpd.conf
```

-m설정값을 사용하면 각각 해당되는 등록부나 파일의 권한도 함께 볼수 있다.

```
kcc@linux:~
# namei -m c/httpd/conf/httpd.conf
f: /etc/httpd/conf/httpd.conf
drwxr-xr-x /
drwxr-xr-x etc
drwxr-xr-x httpd
drwxr-xr-x conf
-rw-r—r— httpd.conf
```

【관련명령문】

ls : 등록부의 내용을 보여주는 명령문

stat : 파일이나 파일체계의 상태를 보여주는 명령문

netstat

/bin/netstat

망상태보기

현재 망의 련결상태와 사용하는 포구 등을 한눈에 확인할수 있게 하는 명령문이다.

형식 : netstat [설정값]

-a, —all : 모든 소케트의 상태를 보여준다.

-c, —continuous : 매 초마다 정보를 갱신하여 계속적으로 보여준다.

-i, —interface : 체계의 망대면부정보를 추가적으로 보여준다.

-n, —numeric : 국부주소와 나가는 주소를 령역형태로가 아니라 IP주소형태로 된 망주소로 보여준다.

-n, —timers : 망작업시계(networking timer)와 관련된 정보를 추가적으로 보여준다.

-p, —program : 매 소케트에 포함된 프로그람의 PID와 이름을 보여준다.

-r, —route : 핵심부경로표를 보여준다.

-s, —statistics : 각 UDP, TCP, ICMP, IP통신규약의 주요통계를 보여준다.

-t, —tcp : TCP소케트만을 보여준다.

-u, —udp : UDP소케트만을 보여준다.

-V, —version : 판본정보를 보여준다.

-h, —help : 도움말을 보여준다.

netstat명령은 체계에 련결된 망과 관련하여 활성화된 정보를 상세하게 볼수 있게 한다. 이것을 아래의 실례들을 통하여 보기로 하자.

netstat -nr설정값으로는 경로표(routing table)를 볼수 있다. 이것은 route명령과 같은 기능을 수행한다.

```
kcc@linux:~

# netstat -nr
Kernel IP routing table
Destination  Gateway     Genmask        Flags MSS  Window  irtt Iface
192.168.0.0  0.0.0.0     255.255.255.0  U      40   0       0   eth1
61.40.233.0  0.0.0.0     255.255.255.0  U      40   0       0   etho
127.0.0.0    0.0.0.0     255.0.0.0      U      40   0       0   lo
0.0.0.0      61.40.233.1 0.0.0.0        UG     40   0       0  eth0
```

아래에 매 출력항목들과 내용설명을 주었다.

출력항목명	설 명
Gateway	경로조종에서의 게트웨이
Genmask	경로조종의 netmask
Flags	경로조종에 관한 여러 기발(Flag)을 표시
Flags	U대면부가 up인 상태
H	경로조종을 통한 호스트에로의 련결
G	게트웨이의 경로
D	경로가 재지정되여 동적으로 생성
M	경로가 ICMP redirect통보문을 통해 수정된 경우
MSS	최대토막크기(Maximum Segment Size)
Window	원격지에서 한번에 보내는 자료의 수신량
irtt	Initial round trip time으로서 0은 초기설정값을 사용한다는 의미
Iface	망대면부

netstat -i 설정값은 매 망대면부장치의 망상태 및 통계를 보여준다.

```
kcc@linux:~

# netstat -i
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OKTX-ERR TX-DRP
TX-OVR Flg
eth0  1500   0 9960659  10   72  10   4725012   0   0  0   BMRU
eth1  1500   0 1254      0    0   0   48        0   0  0   BMRU
lo    16436  0  6291     0    0   0   6291      0   0  0   LRU
```

Eth0, eth1, IO 매 대면부의 MTU(Maximum Transfer Unit)값, 송수신파케트의 오유, 실행초과(Over Run) 등을 볼수 있다.

아래에 매 출력항목들과 내용설명을 주었다.

출력항목명	설　　명
Iface	망대면부
MTU	최대전송단위(Maxium Transmission Unit)
Met	메트리크값
RX-OK/RX-ERR	파케트수신/파케트수신오유
RX-DRP/RX-OVR	파케트수신손실/파케트수신초과
TX-OK/TX-ERR	파케트송신/파케트송신오유
TX-DRP/TX-OVR	파케트송신손실/파케트송신초과

netstat -ta 설정값은 TCP소케트와 관련된 모든 봉사기의 목록을 보여준다.

```
# netstart -ta
```

netstat -anp 설정값은 흔히 망련결상태를 확인하는데 많이 리용된다.

```
# netstat -anp | more
Active Internet connections (severs and established)
Proto Recv-Q Send-Q Local Address        Foreign Address
State   PID/Program name
tcp    0   0 0.0.0.0:515     0.0.0.0:*   LISTEN     28019/lpd Waiting
tcp    0   0 0.0.0.0:3306    0.0.0.0:*   LSTEN      25239/mysqld
tcp    0   0 0.0.0,0:110     0.0.0.0:*   LISTEN     2691/xinetd
```

Program name에 대한 매 상태(state)가 LISTEN이면 이 체계에서 봉사되고있는 망봉사로 볼수 있다. 여기에서는 lpd, mysqld, xinetd데몬의 상태가 LISTEN으로서 봉사상태이며 매 봉사포구는 Send-Q항목의 값인 515, 3306, 110과 같다.

newgrp /usr/bin/newgrp

집단변경

자기가 소속된 집단을 새 집단으로 바꾸는 명령문이다.

형식 : newgrp[집단]

newgrp명령은 현재의 session사용자집단을 변경할수 있다. 새로운 파일을 사용자의 집단이 아닌 다른 집단으로 만들려고 하는 경우, 파일생성후에 chown으로 사용하려는 집단을 지정하여야 하지만 newgrp를 사용하면 변경된 집단속성으로 파일이 생성된다. newgrp에서 지정하는 집단은 현재 존재하는 집단이름이나 GID이여야 한다.

현재 존재하는 집단이름이나 GID는 /etc/group파일에서 확인할수 있다.

touch명령으로 파일을 생성한 경우에 소유자의 기본집단대로 파일의 집단이 정해진다. newgrptest1파일을 touch명령으로 생성해보자.

[주의]
GID 값은 /etc/group을 참고하자.

```
# touch newgrptest1
```

newgrptest1파일은 소유자뿌리에 root집단으로 생성되였다.

```
kcc@linux:~
# ls -al newgrptest 1
-rw-r—r--    1 root    root    0  6월 6 21:36 newgrptest
```

newgrp명령으로 현재의 사용자집단을 admin으로 변경한 다음에 다시 파일을 생성해보자.

```
# newgrp admin
# touch newgrptest2
```

```
kcc@linux:~
# ls -al  newgrptest2
-rw-r—r--    1 root admin        0   6월 6 21:37 newgrptest2
```

현재 사용자의 집단이 admin으로 변경되였으므로 새로 만든 파일의 집단 또한 admin집단에 속하게 된다.

참 고

UNIX

UNIX의 기원은 1969년에 벨연구소의 켄 톰슨과 데니스 리치가 개발한 시분할체계로부터 시작되였다.

UNIX라는 이름은 그 이전의 체계인 MULTICS(Multiplexed Information and computing System)에 기초하여 명명된것으로 알려지고있다. 1974년에 UNIX는 C언어에 의해 다시 만들어졌으며 상업적성격을 띤 여러 개발제공단위들과 대학, 그리고 개인에 의해 많은 확장판이 생겨나고 다양한 판본의 UNIX가 공존하게 되였다.

UNIX는 특정한 콤퓨터회사가 독점하는 조작체계가 아니였으며 표준프로그람작성언어로 작성되였고 대중적인 설계구상을 받아들여 개선해나가면서 최초의 개방형표준조작체계로 발전하였다. 다양한 판본의 UNIX로부터 나온 C언어와 쉘대면부의 복합체는 IEEE의 참가하에 POSIX로 표준화되였다.

POSIX대면부들은 번갈아가며 단일한 UNIX규격에 명시되였다. UNIX의 공식적인 등록상표는 현재 산업표준기구인 Open Group이 소유하고있으며 UNIX관련제품들의 인증과 상표부여 등을 진행하고있다.

UNIX는 Sun Mycrosystems, Silicon Graphics, IBM회사를 비롯한 많은 회사들에서 개발제작하여 판매하고있는 전용작업기(Workstation)들에 적재되여 쓰이고있으며 UNIX의 한 갈래로 널리 쓰이고있는 BSD(Berkeley Software Distribution)도 역시 전용작업기와 일반 PC들에서 적지 않게 사용되고있다.

UNIX환경과 의뢰기/봉사기프로그람방식은 인터네트와 망중심의 새로운 콤퓨터활용을 등장시키고 발전할수 있게 한 중요한 요소로 되였다. 또한 UNIX로부터 파생되여 현재 무료쏘프트웨어와 상업용배포판이 함께 존재하고있는 Linux도 역시 장래의 독점적지위를 차지하게 될 조작체계로 크게 주목되고있다.

newusers /usr/bin/newusers

사용자의 추가

파일을 리용하여 사용자를 추가하거나 사용자정보를 갱신하는 명령문이다.

형식 : newusers 파일

newusers는 체계명령문으로서 /etc/passwd파일과 같은 형식의 파일로부터 사용자를 추가하거나 사용자정보를 갱신한다. 그러나 pwconv명령으로 /etc/shadow파일이 생성된 경우, 이 명령은 수행되지 않는다. 현재 사용되고있는 거의 모든 체계들에서는 보안을 위해 shadow파일을 사용하므로 newusers는 거의나 리용되지 않는다.

참 고

shadow

adduser명령으로 식별이름을 추가하는 경우에 /etc/passwd파일에 사용자이름과 ID, 암호화된 통과암호, 쉘, 사용자 home등록부 등의 정보가 보관된다. 여러 사용자가 리용하는 체계의 경우, 모든 사용자가 /etc/passwd 파일을 읽을수 있어야 체계접속(login)이 가능하게 되므로 passwd파일은 모두에게 공개된것이나 다름없다.

이러한 문제점을 해결하기 위해 만들어진것이 바로 pwconv(shadow)이다.

pwconv는 암호화된 통과암호만을 /etc/shadow파일에 넣어두고 권한을 막아놓아 아무 사용자나 암호를 볼수 없도록 만들어 놓았다.

nice

/bin/nice

우선권순위변경

일정작성의 우선권을 변경하여 프로그람이 동작할 때에 할당되는 순위를 변경한다.

형식 : nice [설정값][명령문][인수]...]

-n 조정수값, -조정수값, —adjustment=조정수값 : 명령의 우선권에 10대신 조정수값을 더한다.

—help : 사용법을 보여준다.

—version : 판본정보를 보여준다.

nice명령으로 프로그람의 우선권을 변경할수 있는데 만약 아무런 인수도 주어지지 않으면 상속받은 현재의 일정작성(scheduling)우선권을 출력한다.

인수가 있는 경우에는 조정된 일정을 먼저 실행한다.

조정수값이 생략되면 명령의 우선권은 10으로 실행하며 nice에 의해 조정될수 있는 범위는 -20(가장 높은 우선권)으로부터 19(가장 낮은 우선권)까지이다. 고급사용자만이 부의(-)조정수값을 부여할수 있다.

nice명령이 프로그람실행시의 우선권순위변경을 지정하는데 비해 renice 명령은 실행된 프로세스가 가지게 될 우선권순위를 지정한다. Nice에서 정수값에는 -를, 부수값에는 --를 사용하지만 renice에서는 정수값에는 -를 사용하지 않는다.

vi를 배경(background)프로쎄스로 2개정도 실행해보자.

```
# vi &
# vi &
```

ps aux명령으로 vi의 프로쎄스ID를 알아보자.

```
kcc@linux:~
# ps aux | grep vi
root   28777 0.0 0.6 2772 856 pts/2   TN  22:08  0:00 vi
root   28780 0.0 0.6 2772 856 ptst/2  TN  22:08  0:00 vi
root   28818 0.0 0.5 2496 688 pts/2   S   22:20  0:00 grep vi
```

그러면 renice명령으로 이 프로그람들의 우선권을 변경해보자.
renice -3명령으로 28777프로쎄스에 우선권순위 -3을 부여한다.

```
kcc@linux:~
# renice -3 28777
28777: old prority 10,  new priority -3
```

renice -2명령으로 28780프로쎄스에는 -2의 우선권순위를 부여한다.

```
kcc@linux:~
# renice -2 28780
28780: old priority 10, new priority -2
```

이 우선권순위를 다시 6의 우선권순위로 변경하는것도 가능하다.

```
kcc@linux:~
# renice 6 28780
28780: old priority-2, new priority 6
```

ps l 28777 28780으로 우선권순위를 한번 확인해보자. 각각 -3과 6의 우선권순위를 가지고있다.

```
kcc@linux:~
# ps l 28777 28780
F UID PID PPID  PRI NI VSZ RSS WCHAN STAT TTY TIME COMMAND
000 0 28777 28567 9 -3 2772 856 do_sig T< pts/2 0:00 vi
000 0 28780 28567 13 6 2772 856 do_sig TN pts/2 0:00 vi
```

nice명령은 실행시의 우선권순위를 지정할수 있다.
7명령으로 7의 우선권순위를 부여해보자.

```
# nice 7 vi &
```

그럼 방금 실행한 vi의 PID를 확인해 보자.

```
kcc@linux:~

# ps aux | grep vi
root 28777 0.0 0.6 2772 856 pts/2 T< 22:08 0:00 vi
root 28780 0.0 0.6 2772 856 pts/2 TN 22:08 0:00 vi
root 29057 0.2 0.6 2768 852 pts/1 TN 22:39 0:00 vi
root 29059 0.0 0.5 2496 688 pts/1 S 22:39 0:00 grep vi
```

ps 1 명령으로 nice가 7이 부여된것을 볼수 있다.

```
kcc@linux:~

# ps l 29057
F   UID PID PPID PRI NI VSZ RSS WCHAN TTY TIME COMMAND
000   0 29057 28952 13 7 2768 852 do_sig TN pts/1 0:00 vi
```

nl

/usr/bin/nl

행번호매기기

본문파일의 매 행에 행번호를 매기는 명령문이다.

형식 : nl[설정값][파일]

-b, —body-numbering=STYLE : 지정한 style을 리용한다. 사용가능한 STYLE은 아래에서 보여준다.

-d, —selection-delimiter=CC : 지정한 CC를 론리페지의 구분으로 한다. 기본은 \: 이다.

-f, —footer-numbering=STYLE : 지정한 STYLE을 번호로 할당할 론리페지꼬리표제행으로 선택한다. 기본값은 n이다.

-h, —header-numbering=STYLE : 지정한 STYLE을 번호를 매길 론리페지머리 표제행으로 선택한다. 기본값은 n이다.

-i, —page-increment=num : 지정한 num만큼 론리페지행번호를 늘인다.

-l, —join-blank-lines=num : 지정한 num값을 공백행수로 하여 번호를 매긴다. -2라면 매 두번째 공백행의 번호를 매기게 된다.

-n, —number-format=FORMAT : 지정한 FORMA를 행번호지정형식으로 사용한다. 사용가능한 FORMAT는 아래설명에 주었다.

-p, —no-renumber : 론리페지분리문자에서 번호매김을 재시작하지 않는다.

-s, —number-separator=string : 지정한 string으로 본문을 행번호와 분리한다.

-v, —first-page=num : 지정한 num값으로 첫번째 론리페지행번호를 설정한다.

-w, —number-width=num : 지정한 num값을 행에 있는 문자의 개수로 사용한다.

—help : 도움말을 보여준다.

—version : 판본정보를 보여준다.

nl명령은 지정한 파일을 읽어 파일의 행번호를 자동적으로 붙여가지고 표준출력으로 보여주는 명령문이다. 이와 같은 기능으로서는 cat -b명령이 있다.

지정가능한 STYLE류형들은 다음과 같다.

- a: 모든 행에 행번호를 출력한다.
- t: 빈 공백란을 제외한 행번호를 출력한다.
- n: 행번호를 출력하지 않는다.
- pREGEXP: 정규표현식에 맞는 행에 한해서만 행번호를 출력한다.

지정가능한 FORMAT류형들은 다음과 같다.

- ln: 0을 제외한 좌측의 맨끝에 행번호를 출력한다.
- rn: 0을 제외한 우측의 맨끝에 행번호를 출력한다.
- rz: 0을 포함한 우측의 맨끝에 행번호를 출력한다.

```
kcc@linux:~

# nl /etc/sendmail.cf | more
   1 #
   2 # Copyright (c) 1998-2001 Sendmail, Inc, and its suppliers.
   3 #       All rights reserved.
   4 # Copyright (c) 1983, 1995 Eric P. Allman. All rights reserved.
   5 # Copyright (c) 1988, 1993
   6 #       The Regents of the University of California All rights reserved.
   7 #
   8 # By using this file, you agree to the terms and conditions
```

[주의]

cat -n[파일]로도 같은 효과를 볼수 있다.

nohup

/usr/bin/nohup

체계탈퇴중에도 프로그람실행

체계탈퇴(log out)를 하는 동안에도 실행중인 명령을 정지하지 않고 수행하게 하는 명령문이다.

형식 : nohup명령[인수..]

nohup는 대용량의 프로그람이나 고화질의 동화상 등을 내리적재받을 때, 또는 프로그람코드작성후의 번역작업 등에 두 세시간정도의 오랜 시간을 기다려야 하는 경우에 효과적으로 사용되는 명령문이다.

```
$ nohup find / -name*.c &>result &
```

nohup는 명령에 대하여 Hangup신호를 무시하면서 우선권을 5만큼 증가시켜 체계탈퇴후에도 계속 배경방식(background)으로 실행하게 한다.

체계탈퇴후에 배경방식에서 명령을 수행하려면 다음과 같이 입력한다.

find같은 명령은 /등록부부터 찾게 되면 매우 오랜 시간이 걸릴수 있다.

nohup명령으로 이것을 체계탈퇴한후에도 실행하도록 할수 있다.

[주의]

xlock와 같은 화면보호프로그람은 화면잠그기기능도 동시에 사용할수 있다. 이것을 사용하는것이 더 편리하기는 하지만 X Window에서 동작하는 어떤 프로그람을 리용하여 다른 사람이 자신의 체계에 접근하지 못하도록 하는것은 위험한 일이다. 장시간 자리를 비워야 하는 경우에는 ouhup을 사용하는것이 낫다.

nslookup /usr/bin/nslookup

이름봉사기문의

령역이름봉사기에 문의하여 결과값을 되돌려받는 명령문이다.

형식 : nslookup[-설정값...][host_to_find|-봉사기]

nslookup은 이름봉사기설정을 검사하는 좋은 명령문이다.

/etc/resolv.conf에 지정되여있는 이름봉사기를 기본값으로 인식하여 문의(query)를 보낸다(이름봉사기의 자세한 설명에 대해서는 데몬설명부분을 참고할것).

```
kcc@linux:~
# cat /etc/resolv.conf
nameserver 61.40.233.122
nameserver 168.126.63.1
```

noninteractive방식으로 찾으려고 하는 호스트의 주소(address)를 문의할수 있다. 만일 아래와 같은 오유통보문이 나오면 이름봉사기설정이 제대로 되지 않은 결과이므로 관련설정을 확인하여야 한다.

```
kcc@linux:~
# nslookup linuxroot.co.kp
**server can't find linuxroot.kp: NXDOMAIN
```

아래와 같이 Name은 linuxroot.co.kp로, Address는 61.40.233.122로서 우의 /etc/resolv.conf의 내용과 동일하게 나오므로 이름봉사기의 설정이 이상없다는것을 알수 있다.

```
kcc@linux:~

# nslookup linux.co.kp
Note: nslookup is deprecated and may  be removed from future
relases.
Consider using the dig' or host' programs instead. Run nslookup
with the -sil[ent]' option to prevent this message from appearing.
Server:   61.40.233.122
Address:   61.40.233.122#53

Name: linuxroot.co.kp
Address: 61.40.233.122
```

Interactive방식으로 DNS기록(record)형식에 대해 자세한 문의를 할수 있다.

이것은 nslookup명령뒤에 나타나는 《>》입력부분에 이름봉사기의 문의를 위한 기록을 set type형식으로 지정하여 자세한 정보를 볼수 있다.

먼저 set type=SOA방식으로 linuxroot.co.kp에 문의를 해보자.

이것을 통해 이름봉사기의 forward zone파일의 SOA기록값들을 확인할수 있다.

```
kcc@linux:~

# nslookup
Note: nslookup is deprecated and may be removed from future
releases.
Consider using the dig' or host' programs instead. Run nslookup
with the -sill[ent]' option to prevent this  message from appearing.
> set type=type
unknown query type: type
> linuxroot.co.kp
Server:   61.40.233.122
Address:   61.40.233.122#53

Non-authoritative answer:
Name: linuxroot.co.kp
Address: 61.40.233.122
> set type=SOA
> linuxroot.co.kp
```

```
Server:    61.40.233.122
Address:    61.40.233.122#53

linuxroot.co.kp

        origin  = ns.linuxroot.co.kp
        mail addr  = root.ns.linuxroot.co.kp
        serial  = 2002031200
        refresh  =28800
        retry  = 14400
        expire  = 3600000
        minimum  = 86400
>
```

set type=MX방식으로 MX기록을 문의해보자. 이름봉사기설정의 mail server는 mail.linuxroot.co.kp로 되여있는것을 확인할수 있다.

```
kcc@linux:~
> set type= MX
> linuxroot.co.kp
Server:   61.40.233.122
Address:   61.40.233.122#53

linuxroot.co.kp mail exchanger  = 10 mail.linuxroot.co.kp.
>
```

type ANY설정값으로 주어진 호스트명과 관련된 모든 자원기록을 볼수 있다. 관련type값은 다음과 같다.

```
kcc@linux:~
> set type=ANY
> linuxroot.co.kp
Server:   61.40.233.122
Address:     61.40.233.122#53
linuxroot.co.kp
Origin  = ns.linuxroot.co.kp
```

```
      mail addr = root.ns.linuxroot.co.kp
      serial = 2002031200
      refresh = 28800
      retry = 14400
      expire=3600000
      minimum = 86400
linuxroot.co.kp nameserver = ns.linuxroot..co.kp.
linuxroot.co.kp mail exchanger=10 mail.linuxroot.co.kp.
Name: linuxroot.co.kp
Address:  61.40.233.122
>
```

set type에 지정가능한 관련 type값들은 다음과 같다.

A	호스트주소
ANY	대부분 관련된 정보
CNAME	alias와 관련된 CNAME(Canonical name)기록정보
HINFO	호스트의 CPU와 조작체계를 지정하는 HINFO기록정보
MD	MD(Mail destination)기록정보
MG	MG(Mail group member)기록정보
MIINFO	Mailbox나 Mail list관련인 MINFO(Mailbox or mail list information)기록정보
MR	MR(Mail rename)기록정보
MX	MX(Mail exchanger)기록정보
NS	NS(Nameserver)기록정보
PTR	호스트이름지정PTR기록정보
SOA	SOA(Domain start-of-authority)기록정보
UINFO	UINFO(User information)기록정보

O

od /usr/bin/od

8진수로 파일보기

파일의 내용을 8진수 또는 다른 형식으로 출력표시(dump)하여 보여준다.

형식 : od[설정값][FILE]...

-a : 바이트를 문자로 표시한다.
-b : 바이트를 부호없는 8진수로 표시한다.
-c : 바이트를 ASCII문자 혹은 Escape문자를 사용하여 표시한다.
-d : 2byte를 부호없는(unsigned) 10진수로 표시한다.
-f : 4byte를 류동소수점수로 표시한다.
-h : 2byte를 16진수로 표시한다.
-i : 2byte를 10진수로 표시한다.
-l : 4byte를 10진수로 표시한다.
-o : 2byte를 8진수로 표시한다.
-x : 2byte를 16진수로 표시한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

파일을 원하는 형식으로 보여준다.
실례로 작은 본문을 2byte씩 16진수와 10진수로 출력해보자.

```
kcc@linux:~

# od -hi test.txt
0000000 696a  6a61 6964 616a  6965 666a 6961 6a65
      26986 27233 26980 24938 26981 26218 26977 27237
000020 6961 666a 696c 0a65
      2697 26218 26988 2661
0000030
```

참 고

출력표시(dump)

주기억의 내용을 출력한다. 주기억장치의 내용을 확인하여 오유수정을 하는데 많이 쓰인다.

P

passwd /usr/bin/passwd

암호의 변경

사용자의 암호를 변경하는 명령문이다.

형식 : passwd[설정값] [사용자]

-l : 지정한 사용자에게 자물쇠를 걸어 더 이상 접근하지 못하게 한다.
-u : -l설정값에 의해 자물쇠가 걸린 사용자에게 자물쇠를 풀어준다.
-d : 사용자의 암호를 삭제한다.
-S : 지정한 사용자의 password 암호화정보를 보여준다.

passwd명령은 사용자의 암호를 변경한다. 체계관리에서 암호는 보안상견지에서 매우 중요하다. 주기적으로 passwd명령을 사용하여 암호를 변경함으로써 보안위험을 최소화할수 있다. 또한 뿌리는 자신뿐만아니라 일반사용자의 암호도 임의로 변경시킬수 있다.

한가지 류의할것은 useradd명령으로 사용자를 추가한후에 반드시 passwd명령으로 암호를 생성해야 한다.

그렇지 않은 경우 추가된 사용자는 체계접속을 할수 없다.

현재 뿌리로 체계접속하였으므로 고급사용자의 암호를 변경하게 된다.

```
kcc@linux:~
# passwd
New password
Retype new password:
passwd: all authentication tokens updated successfully
```

super(특권을 가진)사용자는 사용자의 암호를 변경할수도 있다.

passwd admin명령으로 admin의 새로운 암호를 변경하여보자.

```
kcc@linux:~

# passwd admin
New password:
Retype new password:
passwd: all authentication tokens updated successfully
```

passwd명령과 밀접한 관련이 있는 /etc/passwd의 형식을 한번 보자.

etc/passwd 는 username:password:uid:gid:gecos:homedir:shell 의 형태로 되어있다.

```
kcc@linux:~

# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
 ---중간 생략---------------
admin:x:500:500:admin:/home/admin:/bin/bash
orcle:x:502:5000::/home/oracle:/bin/bash
```

매 출력항목들은 다음과 같다.

username	고유한 식별이름을 구별하는 사용자ID
password	사용자의 password가 암호화된 형태로서 사용자가 읽을수 있다. 보안위험이 있으므로 pwconv 명령문으로 /etc/shadow파일을 사용한다.
uid	사용자ID로써 식별이름을 구별하는 값이다.
gid	집단ID로써 /etc/group과 관련된다.
gecos	사용자의 실제이름, 주소, 전화번호 등의 정보가 있다.
homedir	사용자의 home등록부
shell	사용자가 체계접속할 때의 쉘

paste

/usr/bin/paste

파일합치기

하나이상의 파일에서 행을 합치는 명령문이다.

형식 : paste[설정값]파일…

-d, --delimiters=char : 지정한 char로 렬(column)지시자를 설정한다.
-s, --serial : 하나의 파일이 가로 일렬로 하여 합치게 한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

paste명령은 지정된 파일들의 해당한 행을 합쳐 하나의 표준출력으로 보여주는 명령문이다.

실례를 통해 한번 보기로 하자.

아래에서 test1파일과 test2파일을 먼저 보자.

```
kcc@linux:~
# cat test1
1111 3333
5555 7777
9999 9999
# cat test2
2222 4444
6666 8888
0000 0000
```

paste명령은 이 2개의 파일들을 하나로 만들어 표준출력으로 보여준다.

```
kcc@linux:~
# paste test1 test2
1111  3333  2222  4444
5555  7777  6666  8888
9999  9999  0000  0000
```

-s설정값을 사용하면 매 파일에 있는 모든 내용이 가로로 출력된다.

```
kcc@linux:~
# paste -s test1 test2
1111  3333  5555  7777  9999 9999
2222   4444  6666  8888  0000 0000
```

아래의 실례에서는 ls명령으로 /에 있는 모든 등록부를 한 렬에 출력하되 paste -d설정값으로《;》를 매 렬에 대한 구분기호로 표시하였다.

```
kcc@linux:~
# ls / | paste -d  ';' -s
bin;boot;dev;etc;hdb1;hdb2;hdc1;home;initrd;lib;lost+found;misc;mnt;o
pt;proc;root;sbin;tftpboot;tmp;usbfd;usr;var
```

참 고

GNOME (GNU Network Object Model Environment)

GNOME은 UNIX기초의 조작체계사용자를 위한 도형방식사용자대면부와 일련의 콤퓨터탁상응용프로그람들이다. 사용자에게 편리하고 익숙한 대면부와 함께 여러 응용프로그람을 함께 제공하여 사용자가 쉽게 UNIX/Linux를 사용할수 있도록 한다.

GNOME은 리챠드스톨맨에 의해 설립된 단체인 FSF가 후원하고있으며 GNOME 프로그람과 다른 환경의 프로그람들이 서로 호환될수 있도록 CORBA를 지원하는 ORB와 함께 나오고있다. GNOME는 또한 프로그람작성자들이 GNOME사용자대면부를 사용하는 응용프로그람들을 개발하는데 사용할수 있도록 위자드서고(Wizard Library)를 포함한다.

pathchk /usr/bin/pathchk

파일이름검사

파일이름이 유효하며 다른 체계에로 이식성이 있는가를 검사하는 명령문이다.

형식 : pathchk[설정값]파일이름

-p, —portability : 현재 파일체계에서 길이검사를 하는것이 아니라 다른 체계에로의 이식가능성을 고려하여 POSIX.1 최소규격에 맞는가를 검사한다.

pathchk명령문은 파일이름이 유효하며 이식성이 있는가를 검사하는 명령문으로서 조건이 만족하는 경우는 다음과 같다.

- 파일이름에 속한 모든 등록부에 검색(실행)허가권을 가지고있다.
- 파일이름의 길이가 파일체계가 지원하는 최대파일길이를 넘지 않는다.
- 존재하는 등록부이름에 대한 파일이름의 매 요소길이가 파일이름부분에 대한 파일체계의 최대길이를 넘지 않는다.

완료상태는 다음과 같다.

- 0: 모든 파일이름이 검사내용에 만족할 때
- 1: 기타

p설정값을 리용하면 검색되는 파일이 POSIX최소규격에 맞는가를 확인해볼수 있다. 아래의 실례와 같이 파일이름이 16문자인것은 POSIX최소규격인 14자를 넘는것으로서 POSIX규격에 맞지 않으므로 다른 체계에 이식하는데서 문제가 발생할수 있다.

```
kcc@linux:~
# pathchk -p root/backup/scripts/wwwdb_connect.sh
pathchk: name wwwdb_connect.sh' has length 16; exceeds limit of 14
```

참 고

본문파일의 행사이에 공백넣기

본문파일중 행사이에 공백이 없는 문서를 읽을 때, 눈피로를 덜어주기 위하여 다음과 같은 방법으로 공백을 넣어줄수 있다.

```
# awk '{ptintf('%s\n:\n"),$0}':in.txt>out.txt
```

이렇게 하면 in.txt파일에 매 행마다 공백이 추가되여 out.txt파일이 만들어진다.

프로그람실행시간보기

프로그람이 얼마동안 실행되는가 하는 정확한 시간을 알려고 할 때에는 다음과 같이 한다. 아래의 례는 핵심부번역과정중의 하나이다.

```
# time -v sh -c 'make zlilo'
```

이식성

현재 체계에서 동작하고있는 프로그람이 다른 체계에서도 부정합이나 무리가 없이 동작한다면 이것을 이식성이 좋다고 말한다.

pidof

/sbin/pidof

PID확인

실행중인 프로그람의 프로세스ID를 찾는 명령문이다.

형식 : pidof [설정값]프로그람

-s : 지정한 프로그람 PID가 여러개 있는 경우에 하나만을 보여준다.
-x : 스크립트와 함께 실행중인 프로그람이면 그 쉘의 id도 함께 보여준다.
-o pids : 찾으려고 하는 프로그람이름중에서 pids로 지정된 PID만 제외하고 나머지를 보여준다.

pidof명령은 프로그람의 프로세스ID를 찾는다.
이것은 프로세스를 만든 명령문은 알지만 해당 PID를 모르는 경우에 매우 유용하게 사용할수 있다.
ps aux명령중 grep명령으로 vi와 관련된 프로세스만을 뽑아보자.
이것은 자세한 vi에 대한 프로세스정보들을 볼수 있다.

```
kcc@linux:~

# ps aux | grep vi
root 28777 0.0 0.6 2772 856 pts/2 TN 22:08 0:00 vi
root 28780 0.0 0.6 2772 856 pts/2 TN 22:08 0:00 vi
root 28818 0.0 0.5 2496 688 pts/2 S  22:20 0:00 grep vi
```

아래와 같이 vi프로세스에 대한 PID만 필요한 경우에도 pidof명령문을 사용할수 있다.

```
kcc@linux:~

# pidof vi
28777 28780
```

이에 대한 보다 자세한 설명에 대해서는 nice명령을 참고할 필요가 있다.

ping

/bin/ping

망련결확인

ICMP ECHO_REQUEST파케트를 망호스트에 전송하여 망련결성을 확인하는 명령문이다.

형식 : ping[설정값]host

-b : broadcast주소로 ping을 보낸다.

-c count : 지정한 count수자만큼 파케트를 보낸후에 끝낸다.

-f : ping Flooding파케트를 보낼수 있다. 이것은 초기의 UNIX체계들에서 DoS ping 공격으로 리용되였다.

-i wait : 지정한 wait간격으로 파케트를 보낸다.

-l interface address : 발신지주소의 대면부주소를 지정한다.

-l preload : 정상적인 방식에서 ping을 보내기전에 지정한 preload수만큼 미리 파케트를 가능하게 고속으로 보낸다.

-n : ping을 보내는 대상의 파케트가 돌아오는 호스트이름을 IP주소형태로 보여준다.

-p pattern : 송신할 파케트를 채우기 위해 16 pad' ' 바이트까지 지정한다. 이것은 망의 자료관련문제를 진단하는데 유용하다. 례를 들어 -p ff는 모두 1로 파케트를 채운다.

-q : ping파케트의 반응상황을 보여주지 않고 통계만 보여준다.

-r : 경로표를 리용하지 않고 직접적으로 련결된 호스트에 파케트를 보낸다.

-s packetsize : 지정한 packetsize만큼의 파케트를 보낸다. 기본값은 56이며 최대값은 65507이다.

-v : ECHO_RESPONSE와 함께 수신된 ICMP파케트의 상세정보를 보여준다.

-V : 판본정보를 보여준다.

ping은 망에 ICMP파케트를 보내여 호스트의 련결성을 분석하는 명령문이다.

망에 련결된 하나의 호스트가 다른 호스트에 ICMP파케트를 보내게 되면 파케트는 대상호스트에 도착한 시간을 기록한 후 다시 파케트를 보낸 호스트로 돌아오게 된다. 이 두 호스트사이에 오고 간 파케트의 왕복시간을 계산하여 망호스트의 련결성이나 호스트의 동작유무를 알수 있다. 만일 응답이 없는 경우에는 파케트를 보낸 호스트의 련결이나 동작에 이상이 발생한것으로 판단할수 있다.

-c설정값으로 보낼 파케트수를 지정한다. 다음 실례에서는 3번을 보내는 설정값을 사용하였다. 파케트왕복시간의 검사에 의해 kp.kcccom.com호스트와의 련결상태가 좋다는것을 확인할수 있다.

```
kcc@linux:~

# ping -c 3 kp.kcccom.com
PING kp.kcccom.com (211.233.3.59) from 61.40.233.122 : 56(84) bytes of data
64 bytes from kp.kcccom. (211.233.3.59): icmp_seq=0 ttf=246 time=5.848 msec
64 bytes from kp.kcccom. (211.233.3.59): icmp_seq=1 ttf=246 time=5.045 msec
64 bytes from kp.kcccom. (211.233.3.59): icmp_seq=2 ttf=246 time=4.918 msec

—— kp.kcccom.com pping statistics———
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev=4.918/5.270/5.848/0.415 ms
```

다음은 61.40.233.125에 파케트를 보내는 경우이다.

되돌아오지 않는것으로 보아 망문제가 있다는것을 예상할수 있다.

```
kcc@linux:~

# ping -c 3 61.40.233.125
PING 61.40.233.125(61.40.233.125)from 61.40.233.122 : 56(84) bytes of data.
From 61.40.233.122: Destination Host Unreachable
From 61.40.233.122: Destination Host Unreachable

—61.40.233.125 ping statistics-
3 packets transmitted, 0 packets received, +3 errors, 100% packet loss
```

[주의]

보안상의 리유로 ICMP파케트의 응답을 금지하는 경우도 있다.

참 고

ping에 대한 응답을 하지 않는 방법

아래와 같이 명령을 입력하면 호스트콤퓨터에 ping반응이 되지 않아 ping명령문으로의 확인이 불가능하다.

```
# echo 1> /proc/sys/net/Ipv4/icmp_echo_ignore_all
```

호스트콤퓨터가 있는 broadcast ping에만 반응을 하지 않을수도 있다.

```
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

이 명령을 다시 기동한후에도 계속 적용하려면 /etc/rc.d/rc.local파일이나 다음 파일에 추가하여 넣는다.

```
kcc@linux:~
# cat /etc/sysctl.conf
net.ipv4.icmp_echo_ignore_broadcasts=1
net.ipv4.icmp_echo_ignore all=1
```

ICMP

인터네트조종통보문규약(Internet Control Message Protocol)으로서 인터네트에서 오유통보문의 생성 및 검사, 그리고 IP에 관련된 정보를 제공하는 통신규약을 말한다. 실례로 ping명령은 인터네트접속검사를 위해 ICMP를 사용한다.

파케트(packet)

망에서 자료전송을 위한 하나의 전송자료단위이다.

파케트크기는 수신측과 전송측에서 사용되는 통신규약에 의하여 결정된다. 일반적으로 128byte가 표준이지만 52, 64, 256Octet 등 편의에 따라 크기를 바꿀수 있다. Octet는 보통 8bit로 구성되며 이것을 1문자로 간주한다.

pr

/usr/bin/pr

파일인쇄설정

본문파일을 인쇄할수 있는 표준출력으로 변환하는 명령문이다.

형식 : pr[파일]

-num_cols, —columns=num=num_cols : 지정한 num_cols값만큼 렬 값으로 설정한다. 기본값은 1이다.

-a, —across : 중복되는 렬을 수직으로 채우지 않고 지정하는 across 값만큼 수평으로 먼저 채운다.

-c, —show-control-chars : 인쇄할 때에 보이지 않는 문자들을 인쇄 가능한 문자로 변환한다.

-d, —double-space : 공백을 double space로 출력한다.

-D, —date-format=FORMAT : 지정한 FORMAT를 header의 날자로 사용한다.

-e[char[width]], —expand-tabs[=char[width]] : tab를 공백문자로 변환한다. 만일 char를 지정하면 지정한 char를 tab문자로 한다(기본 tab 문자). 만일 width값을 지정하면 지정한 값을 tab문자크기로 설정한다(기본값 8).

-F, -f, —form-feed : 새로운 페지에 출력하기 위한 개행문자대신에 인쇄용지이송(form feed)문자를 사용한다.

-h header, —header=header : 지정한 header로 머리부문자렬을 설정한다(기본값 8).

-i[char[width]], —output-tabs[=char[width]] : 공백을 tab로 바꾼다. 만일 char를 지정하면 지정한 char를 tab문자로 한다(기본tab문자). 만일 widdth값을 지정하면 지정한 값을 tab문자크기로 설정한다(기본값 8).

-l page_length, —length=page_length : 페지길이를 지정한 page_length값으로 설정한다. 기본값은 66행이다.

-m, —merge : 모든 파일을 렬당 하나의 파일로 보여준다.

-n[delimiter[digits], —number-linest[=delimiter[digits] : 행번호를 붙인다(기본값은 5). 만일 delimiter가 지정되면 식별자를 지정된 값으로 설정한다(기본값은 tab). 만일 width값이 지정되면 지정된 자리수를 기준으로 행번호를 붙인다(기본값은 5).

-o width, —indent=width : 지정한 width값으로 각 행을 들여쓰기 한다(기본값은 0).
-r, —no-file-warning : 입력된 파일을 열수 없는 경우에도 경고통보문을 보이지 않는다.
-s[delimiter], -separator[=delimiter] : delimiter로 지정된 단일문자로 렬을 구분한다(기본값은 tab)
-t, —omit-header : 페지의 header와 꼬리표제를 생략한다.
-T, —show-nonprinting : -t설정값과 비슷하지만 용지이송(form feed)문자까지 제거한다.
-v, —show-nonprtnting : 인쇄되지 않는 문자를 인쇄가능한 문자로 변환한다.
-w page_width, —width=page_width : 지정한 page_width값으로 페지행을 설정한다(기본값은 72).
-W page_width, —page-width=page_width : 지정한 page_width 값으로 페지행설정을 바꾼다(기본값은 72).
—help : 도움말을 보여준다.
—version : 판본정보를 보여준다.

pr명령문은 본문파일을 인쇄가능한 형태로 페지를 나눈다. 여기에는 페지번호, 날자, 시간 및 파일이름을 가진 머리표제(header)가 포함된다.

아래 실례와 같이 pr명령으로 /etc/sendmail.cf파일을 인쇄가능한 형태로 변경해보자. 페지의 header부분에 인쇄날자와 본문파일이름, 페지번호 등이 기입된것(강조체로 표시됨)을 볼수 있다.

```
kcc@linux:~
# pr /etc/sendmail.cf | more
2002-07-04 18:07   /etc/sendmail.cf Page 1
#
# Copyright (c) 1998-2001 Sendmail, Inc. and its suppliers.
#      All rights reserved
# Copyright (c) 1983, 1995 Eric P. Allman. All rights reserved.
# Copyright (c)  1988, 1993
#      The Regents of the University of California. All rights reserved.
#
```

pr파일이름은 화면에 내용을 보여주므로 파이프로 lpr명령을 사용하여 직접 인쇄를 실행한다.

```
# pr test.txt | lpr
```

-h설정값은 text.txt에 제목을 지정하여 인쇄가 가능하게 한다.

```
# pr -h "TEST PAGES" test.txt |lpr
```

하나이상의 파일을 하나의 인쇄용지에 인쇄할수 있다.

```
# pr -m -h "TEST PAGES" test.txt text2.txt | lp
```

참 고

포스트스크립트(postscript)

포스트스크립트는 인쇄된 페지의 모양을 지정하는 언어로서 1985년에 Adobe회사가 개발하여 인쇄분야의 산업계표준으로 되였다. 주요한 인쇄기제작단위들에서는 기본 조작체계환경에서 실행될수 있는 포스트스크립트 쏘프트웨어가 들어있거나 또는 적재될수 있도록 인쇄기를 제작하여 판매하고있다. 포스트스크립트파일은 확장자가 .ps로 끝나므로 쉽게 분간할수 있다. 또한 포스트스크립트파일은 Adobe PDF형식으로 쉽게 변환하여 사용할수 있다.

praliases

/usr/sbin/praliases

우편 alias보기

특정 식별이름으로 도착하는 우편을 다른 사람에게 넘겨주는 우편alias의 설정내용을 확인하는 명령문이다.

형식 : praliases[설정값]

-C file : 지정한 파일에서 우편설정파일을 읽는다. 기본값은 /etc/sendmail.cf이다.
-f file : 지정한 파일에서 우편별명파일을 읽는다. 기본값은 /etc/aliases 이다.

우편봉사기로 가장 많이 사용되는것이 sendmail봉사기인데 이것은 인터네트전자우편의 표준규약인 SMTP(Simple Mail Transfer Protocol)를 통해 우편봉사를 한다. 우편봉사중의 우편alias기능에 대해 우편사용자의 별명이라고 생각하면 된다. 만일 한 우편사용자가 admin식별이름과 webmaster식별이름, webmin식별이름 등 여러가지를 관리하는 경우, 개개의 우편사용자를 생성한는것보다 개개를 하나의 식별이름으로 alias시키는것이 훨씬 낫다. 이러한 alias된 내용을 볼수 있는것이 praliases명령문이다.

아래 목록과 같이 /etc/aliases파일은 우편alias를 위한 설정파일이다.

여기에서는 postmaster, daemon, adm 등의 식별이름으로 오는 우편을 모두 뿌리식별이름에서 받아볼수 있게 하였다.

```
kcc@linux:~

# praliases
postmastr:root
daemon:root
adm:root
lp:root
shutdown:root
operator:root
ftp:root
apache:root
```

printenv /usr/bin/printenv

환경변수보기

환경변수의 값을 보여주는 명령문이다.

형식 : printenv[이름]

printenv명령은 환경변수들의 값을 보여준다. 이름을 지정하면 지정한 환경변수의 내용만을 보여주며 지정하지 않으면 모든 환경변수의 값을 보여준다.

```
kcc@linux:~
# printenv
PWD=/root
HOSTNAME=ns.linuxroot.co.kp
LD_LIBRARY_PATH=/usr/lib/qt/lib:
PVM_RSH=/usr/bin/rsh
QTDIR=/usr/lib/qt
LESSOPEN=|/usr/bin/lesspipe.sh %s
MANPATH=/usr/lib/qt/man:
XPVM_ROOT=/usr/share/pvm3/xpvm
KDEDIR=/usr
USER=root
```

export명령에 아무런 설정값도 지정하지 않거나 -p설정값을 주면 쉘에서 export된 모든 환경변수의 목록을 보여준다.

```
kcc@linux:~
# export
declare -x BASH_ENV="/root/.bashrc"
declare -x HISTSIZE="1000"
declare -x HOME="/root"
declare -x HOSTNAME="ns.linuxroot.co.kp"
----------생략---------
```

ps /bin/ps

프로세스상태보기

프로세스의 현재 상태를 보여주는 명령문이다.

형식 : ps [설정값]

전체적인 프로세스와 관련된 설정값

-A : 모든 프로세스를 보여준다.

-N : -A설정값과 비슷하나 ps지령을 위해 실행한 ps프로세스만 제외하여 보여준다(실행을 취소한다).

-a : session leader 및 말단과 관련되지 않은 프로세스를 제외한 모든 프로세스를 보여준다.

-d : session leader를 제외한 모든 프로세스를 보여준다.

-e : 핵심부 프로세스를 제외한 모든 프로세스를 보여준다.

T : 현재 말단에서의 모든 프로세스를 보여준다.

a : 한 말단의 사용자고유프로세스를 보여준다.

r : 현재 실행중인 프로세스를 보여준다.

x : 말단없는 프로세스를 보여준다.

—deselect : -N설정값과 같다.

특정프로세스를 선택하여 지정된 목록만 보여주는 설정값

-C : 지정한 명령이름과 관련된 정보를 보여준다.

-G : 지정한 RGID과 관련된 정보를 보여준다(이름도 지원).

-U : 지정한 RUID과 관련된 정보를 보여준다(이름도 지원).

-g : 지정한 session leader 혹은 집단이름에 관련된 정보를 보여준다.

-p : PID를 선택한다.

-s : session에 속한 프로세스를 선택한다.

-t : tty를 선택한다.

-u : 사용자ID를 지정한다(이름도 지원).

U : 지정한 사용자의 프로세스를 보여준다.

P : 프로세스ID를 선택한다.

t : tty를 선택한다.

—Group : 실제 집단이름이나 ID를 선택한다.

—User : 실제 사용자이름이나 ID를 선택한다.
—group : 유효사용자이름이나 ID를 선택한다.
—pid : 프로세스 ID를 선택한다.
—sid : session ID를 선택한다.
—tty : 말단을 선택한다.
—user : 유효사용자이름이나 ID를 선택한다.
-123 : —sid의 의미이다.
123 : —pid의 의미이다.

결과로써 출력하는 항목을 선택(조종)하는 설정값

-O : PID, TTY, STAT, TIME, COMMAND 등의 항목목록을 보여준다.
-c : PID, CLS, PRI, TTY, TIME, CMD 등의 항목목록을 보여준다.
-f : UID, PID, PPID, C, STIME, TTY, TIME, CMD 등의 항목목록으로 CMD항목의 전체 명령문형태를 보여준다.
-j : PID, PGID, SID, TTY, TIME CMD 등의 항목목록을 보여준다.
-l : F, S, UID, PID, PPID, C, PRI, NI, ADDR, SZ, WCHAN, TTY, TIME, CMD 등의 항목목록으로 상세한 정보를 보여준다.
-o : 사용자지정형식
-y : -l이나 l설정값과 함께 쓰이여 ADDR항목을 RSS항목으로 보여준다.
O : PID, TTY, STAT, IME, COMMAND 등의 항목목록을 보여준다.
X : PID STACKP, ESP, EIP, TMOUT, ALARM, STAT, TTY, TIME, COMMAND 등의 항목목록으로 Linux i386 Resistor형식을 보여준다.
j : PPID, PID, PGID, SID, TTY, TPGID, STAT, UID, TIME, COMMAND 등의 항목목록으로 작업조종에 관련된 형식을 보여준다.
l : F, S, UID, PID, PPID, C, PRI, NI, ADDR, SZ, PSS WCHAN, TTY, TIME, CMD 등의 항목목록으로 -l설정값에 PSS항목을 추가하여 보여준다.
o : 사용자지정형식
s : UID, PID, PENDING, BLOCKED, IGNORED, CAUGHT, STAT, TTY, TIME, COMMAND 등의 항목목록을 보여준다.
u : USER, PID, %CPPU, %MEM, VSZ, RSS, TTY, STAT, START, TIME, COMMAND 등의 항목목록을 보여준다.

v : PID, TTY, STAT, TIME, MAJFL, TRS, DRS, %MEM, COMMAND 등의 항목목록을 보여준다.

-format : 사용자지정형식

결과로써 출력하는 항목의 내용을 변경하는 설정값

-H : 프로세스를 계층형으로 보여준다.

-m : 쓰레드(thread)를 보여준다.

-n : 지정한 체계이름목록파일을 사용한다.

-w : 항목의 너비에 맞추어 잘리운 내용을 맞추어 보다 자세하게 보여준다.

C : raw CPU시간을 리용한다.

N : 지정한 체계이름목록파일을 사용한다.

O : 정렬순서지정의 설정값을 사용하여 정렬방식을 O[+l-]K[,[+l-]K[,...]의 형식으로 지정한다. 여기에서의 K는 사용법에서 설명한다. +는 오름순서정렬이며 -는 내림순서정렬이다.

S : 비활성화된 자식프로쎄스자료를 포함한다.

c : 체계내부에 보관된 간단한 지령이름을 보여준다.

e : 지령에 대한 매개변수와 함께 환경변수도 보여준다.

f : ASCII Art로 프로쎄스나무(process tree)를 보여준다.

h : 머리부표제행(header line)을 보여주지 않는다.

m : 모든 쓰레드를 보여준다.

n : WCHAN과 USER항목을 수자값으로 보여준다.

w : 항목의 너비에 맞추어 잘리운 내용을 맞추어 보다 자세하게 보여준다.

—cols : 화면(Screen)의 너비를 설정한다.

—columns : 화면의 너비를 설정한다.

—cumulative : 죽은 자식 프로쎄스자료를 포함한다.

—forest : ASCII Art의 프로쎄스나무를 보여준다.

—html : HTML Escape로 보여준다.

—headers : 머리부표제행(header line)을 반복한다.

—no headers : 머리부표제(header)를 보여주지 않는다.

—lines : 화면에 나오는 행수를 설정한다.

—rows : 화면의 높이를 설정한다.

—sort : 지정한 정렬방식으로 보여준다.

—sort=[+|-]key[,[+|-]key[,...] 형식이다. 여기서 사용할수 있는 key는 사용법에서 설명한다. 실례를 들어 ps jax —sort=uid,-ppid,+pid 형식으로 할수 있다.

프로그람의 정보

-V : 판본정보를 보여준다.

L : 모든 형식지시자를 보여준다.

V : 판본정보를 보여준다.

—help : 도움말을 보여준다.

—info : 오유수정정보를 보여준다.

—version : 판본정보를 보여준다.

[주의]

설정값을 보면 풀이표(-)가 없는것, 하나 또는 두개가 있는것들이 있다. Unix98의 설정값은 풀이표가 하나이며 BSD계렬의 설정값은 풀이표가 없다. 또한 GNU는 두개의 풀이표가 있는 설정값이나 이것들은 모두 호환된다.

ps명령문은 프로세스의 현재 상태를 보여주는 명령문이다.

ps명령으로 현재 사용하는 프로세스의 상태를 간단히 보자.

```
kcc@linux:~

$ ps
  PID  TTY      TIME   CMD
 3177  pts/4  00:00:00  bash
 3599  pts/4  00:00:00  ps
```

ps명령으로 PID, TTY, TIME, CMD header의 항목과 내용을 보여준다. 그러면 이 항목들에 대해 보기로 하자. ()안의 내용은 설정값에 해당한다.

- ADDR: 프로세스탄창(process stack)의 토막번호(-l, l설정값)
- BND: 핵심부 쓰레드가 결합(bind)되는 프로세스의 론리프로세스번호(-O설정값)
- C: (-f,l,-l설정값)

- CMD: 사용자가 실행한 명령이름(-f,-l,-l설정값)
- COMMAND: 사용자가 실행한 명령이름(s,u,v설정값)
- F: 프로쎄스 및 쓰레드에 관련된 항목(-l,l설정값)
- LIM: 주기억기에 대한 쏘프트웨어 한계와 관련된 항목(v설정값)
- NI: 프로쎄스의 nice값, 낮을수록 CPU시간이 높다. (-l,l설정값)
- PID: 프로쎄스ID
- PRI: 프로쎄스일정작성(process scheduling)의 우선권순위, 낮을수록 우선권순위가 높다.(-l,l설정값)
- RSS: 프로쎄스의 실지 주기억기의 크기로서 단위는 KB(-l,l설정값)
- S: 프로쎄스나 핵심부 쓰레드의 상태(-l,l설정값)
- SIZE: 가상환경변수(virtual image)의 크기(v설정값)
- STAT: 실행되고있는 프로쎄스의 상태(s,u,v설정값)

아래에 STAT항목의 상태에 대한 설명내용을 주었다.

-D : 디스크입출력대기상태로서 새치기(interrupts)를 걸수 없는 상태
-R : 실행중
-S : 짧은 대기(sleep)상태
-T : 정지상태
-Z : 비활성(zombie)상태
-W : 상주한 페지가 없는 프로쎄스
-< : 높은 우선권의 프로쎄스
-N : 낮은 우선권의 프로쎄스
-L : 페지에 자물쇠가 걸린 상태

- STIME : 프로쎄스의 시작시간(-f,u설정값)
- SZ : 프로쎄스가 사용하는 자료와 탄창(stack)의 크기 (-l,l설정값)
- TIME : 프로쎄스가 소비한 총시간(기본항목)
- TRS : 본문의 실지 주기억기의 크기(v설정값)
- TTY : 련결되여있는 말단(기본항목)
- UID : 사용자ID (-f,-l,l설정값)

- USER : 사용자이름(u설정값)
- WCHAN : 프로세스에 거주하는 핵심부함수(-l,l설정값)
- VSZ : 가상주기억기에 적재된 프로세스의 크기(단위는 KB)
- %CPU : 마지막 1분동안 프로세스가 사용한 CPU점유률(u설정값)
- %MEM : 마지막 1분동안 프로세스가 사용한 주기억기의 점유률(u,v설정값)

-u설정값으로 사용자의 ID나 이름을 지정하여 지정한 사용자의 프로세스만을 볼수 있다.

kcc@linux:~

```
# ps -u apache -l
  F S UID PID  PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
140 S 48  835  783  0  69  0 -20098  semop ?  00:00:06 httpd
140 S 48  836  783  0  69  0 -20095  semop ?  00:00:00 httpd
140 S 48  837  783  0  69  0  -20097  semop ?  00:00:00 httpd
140 S 48  838  783  0  69  0  -20097  semop ?  00:00:05 httpd
140 S 48  839  783  0  69  0  -20097  semop ?  00:00:07 httpd
140 S 48  840  783  0  69  0  -20097  do_sel ?  00:00:07 httpd
140 S 48  841  783  0  69  0  -20097  semop ?  00:00:09 httpd
140 S 48  842  783  0  69  0  -20098  semop ?  00:00:20 httpd
140 S 48  3864 783  0  69  0  -20095  semop ?  00:00:00 httpd
140 S 48  3865 783  0  69  0  -20092  semop ?  00:00:00 httpd
140 S 48  3866 783  0  69  0  -20095  semop ?  00:00:00 httpd
```

우의 apache관련 프로세스들가운데서 PID 835를 강제로 완료시키자. 여기에서 -9의 입력은 해당한 프로세스를 강제로 완료시킨다.

[주의]
-9는 KILL signal을 의미한다.

```
# kill -9 835
```

관련 PID프로세스를 다시 시작할수도 있다.

```
# kill -HUP 835
```

또는 아래와 같은 방법을 사용할수도 있다.

```
# kill -SIGHUP 835
```

보통 체계의 모든 프로세스를 보려고 하는 경우에는 aux설정값의 조합을 많이 사용한다.

```
kcc@linux:~

$ ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root        1  0.0  0.8  1384  512  ?      S    Jul05   0:03 init [3]
root        2  0.0  0.0     0     0 ?      SW   Jul05   0:00 [keventd]
root        3  0.0  0.0     0     0 ?      SW   Jul05   0:00 [kapm-idled]
root          4   0.0   0.0        0      0?          SWN   Jul05   0:00
[ksoftirqd_CPU0]
root        5  0.0  0.0     0     0 ?      SW   Jul05   0:07 [kswapd]
root        6  0.0  0.0     0     0 ?      SW   Jul05   0:00 [kreclaimd]
root        7  0.0  0.0     0     0 ?      SW   Jul05   0:00 [bdflush]
root        8  0.0  0.0     0     0 ?      SW   Jul05   0:00 [kupdated]
root        9  0.0  0.0     0     0 ?      SW   Jul05 0:00 [mdrecoveryd]
root       13  0.0  0.0     0     0 ?      SW   Jul05   0:00[kjournaid]
root       70  0.0  0.0     0     0 ?      SW   Jul05   0:00[khubd]
------생략-------
```

많은 프로세스를 어떤 기준에 의해 정렬하는 경우가 많다. 정렬을 위해서는 O설정값이나 --sort 설정값을 사용한다. O설정값인 경우, O[+|-]K[,[+|-]K[,...]]의 형식을 사용하며 -sort설정값인 경우에는 --sort=[+|-]key[,[+|-]K[,...]의 형식을 사용한다.

O설정값에 -r를 첨부하여줌으로서 RSS를 가장 큰 값을 맨 우에 보여주는 내림순서로 정렬하였다.

만일 2개이상의 기준으로 정렬을 할 때에는 반점(,)다음에 정렬방식과 기준되는 K항목을 지정하면 된다.

```
kcc@linux:~
$ ps aux O-r
USER       PID  %CPU  %MEM  VSZ  RSS  TTY  STAT  START  TIME
COMMAND
apache  842   0.0 11.0 80392  6816 ?  S  Jul05  0:20  /usr/sbin/httpd-
apache  838   0.0 11.0 80388  6812 ?  S  Jul05  0:05  /usr/sbin/httpd-
apache  839   0.0 11.0 80388  6808 ?  S  Jul05  0:07  /usr/sbin/httpd-
apache  837   0.0 11.0 80388  6804 ?  S  Jul05  0:00  /usr/sbin/httpd-
apache  835   0.0 11.0 80392  6800 ?  S  Jul05  0:06  /usr/sbin/httpd-
apache  840   0.0 11.0 80388  6800 ?  S  Jul05  0:07  /usr/sbin/httpd-
apache  841   0.0 11.0 80388  6796 ?  S  Jul05  0:09  /usr/sbin/httpd-
apache  3864  0.0 11.0 80380  6788 ?  S  08:15  0:00  /usr/sbin/httpd-
apache  3866  0.0 11.0 80380  6788 ?  S  08:15  0:00  /usr/sbin/httpd-
apache  8865  0.0 11.0 80368  6784 ?  S  08:15  0:00  /usr/sbin/httpd-
apache  836   0.0 10.9 80380  6760 ?  S  Jul05  0:00  /usr/sbin/httpd-
root    783   0.0 10.7 80296  6632 ?  S  Jul05  0:00  /usr/sbin/httpd-
mysql   774   0.0 7.2  28208  4484 ?  S  Jul05  0:00  /usr/libexec/mysq
mysql   779   0.0 7.2  80208  4484 ?  S  Jul05  0:00  /usr/libexec/mysq
mysql   780   0.0 7.2  28208  4484 ?  S  Jul05 0:00  /usr/libexeapache
mysql   781   0.0 7.2  28208  4484 ?  S  Jul05  0:00  /usr/libexec/mysq
```

여기에서는 O설정값이나 -sort에서 사용가능한 키이다. K항목은 O설정값에서, key항목은 -sort설정값에 사용한다.

K	key	설 명
c	cmd	실행이름
C	cmdline	전체 명령행
f	flags	기발
g	pgrp	프로세스집단ID
G	tpgid	tty에 속한 프로세스집단ID
j	cutime	루적사용자시간
J	cstime	루적체계시간
k	utime	사용자시간
K	stime	체계시간

m	min_ftt	보다 작은 페지의 기정값
M	maj_ftt	보다 큰 페지의 기정값
n	cmin_ftt	보다 작은 페지의 기정루적수
N	cmaj_ftt	보다 큰 페지의 기정루적수
O	session	Session ID
p	pid	프로쎄스ID
P	ppid	부모프로쎄스ID
r	rss	상주주기억기의 크기
R	resident	상주페지
s	size	KB단위의 사용주기억기의 크기
S	share	공유페지의 수
t	tty	Tty
T	start_time	프로쎄스가 시작된 시간
u	uid	사용자 ID
U	user	사용자이름
V	vsize	byte단위의 전체 상주기억기(VM)의 크기

【관련명령문】

pstree : 나무(tree)형태로 프로쎄스계층도를 보여준다.

top : 실시간으로 프로쎄스변화상태를 보여준다.

참 고

IEEE (Institute of Electrical and Electronics Engineers)

IEEE는 어느 한 나라의 전기전자학회를 말한다. IEEE표준의 개발을 추진하고있으며 이러한 표준들은 많은 경우에 국가표준 및 세계표준으로 되고있다. 많은 종류의 학회잡지들을 발행하고있으며 많은 지역들에 지부를 두고있다. 그리고 여기에는 《IEEE 콤퓨터학회》등과 같이 특정한 분야에 관하여 여러개의 큰 학회들이 부속되여있다.

pstree /usr/bin/pstree

프로세스상관도

프로세스의 부모자식관계를 나무형태로 보여주는 명령문이다.

형식 : pstree[설정값][pid | user]
pstree -V

-a : 지령행에서 지정한 인수가 있는 경우, 지령행인수까지 보여준다.
-c : 기본값은 동일한 나무(tree)안의 같은 프로세스에 관해 하나의 프로세스만 보여주고 해당 프로세스의 개수를 나타내는데 같은 프로세스를 모두 보여준다.
-G : 나무형태를 보기 편리한 VT100형태로 보여준다.
-h : 현재 프로세스와 부모프로세스를 강조체로 보여준다.
-H pid : pid로 지정된 프로세스와 부모프로세스를 강조체로 보여준다.
-l : 긴 행을 모두 보여준다.
-n : PID를 기준으로 정렬하여 보여준다.
-p : PID까지 보여준다.
-u : UID를 보여준다.
-U : UTF-8 (Unicode)형태로 보여준다.
-V : 판본정보를 출력한다.
pid : 지정하는 pid를 보여준다.
user : 지정하는 사용자만의 프로세스를 보여준다.

pstree명령문은 프로세스간의 부모자식관계를 나무(tree)형태로 보여주어 리해하기 쉽도록 한다. 실례로 아래의 화면에서 gaim은 kdeinit의 자식프로세스이라는것, 또한 모든 프로세스는 init의 자식프로세스로 련결되여있다는것을 알수 있다.

```
kcc@linux:~

#pstree
init┼alarmd
  |-bdflush
  |-crond
  |-eth0
  |-eth1
  |-httpd---8*[httpd]
  |-kapm-idled
  |-kdeinit┼kdeinit
  |        -wmami
  |-5*[kdeinit]
  |-kdeinit---gaim
  |-kdm┼X
  |         -kdm---startkde---ksmserver
  |-keventd
  |-khubd
  |-2*[kjournald]
  |-klogd
  |-kreclaimd
  |-ksoftirqd_CPU0
  |-kswapd
  |-kupdated
  |-lpd
  |-mdrecoveryd
  |-6*[mingetty]
  |-named--named--3*[named]
  |-safe_mysqld--mysqld--mysqld--2*[mysqld]
  |-2*[smbmount]
  |-sshd--sshd--bash---pstree
  |--syslogd
  -xfs
```

pwck

/usr/bin/pwck

암호파일검사

암호파일의 보안문제점을 확인하는 명령문이다.

형식 : pwck

pwck명령문은 /etc/passwd파일의 내용을 분석하여 잘못된 내용이 있는가를 검사한다. 이것은 암호의 보안문제점을 미리 발견하는 명령으로서 정기적으로 검사해보는것이 좋다.

이 명령문은 아래에서 보여주는 바와 같이 /etc/passwd파일의 여러 문제점들을 자동적으로 검사하여준다.

```
kcc@linux:~

# pwck
user adm: directory /var/adm does not exist
usr news: directory /var/spool/news does not exist
usr uucp: directory /var/spool/uucp does not exist
usr gopher: directory var/gopher does not exist
usr wnn: directory /home/wnn does not exist
user gdm: directory /home/gdm daes not exist
user pcap: program /bin/nologin does not exist
user junkbust: directory /etc/junkbuster does not exist
pwck: no changes
```

adm식별이름은 /etc/passwd파일에는 home등록부를 /var/adm으로 지정하였지만 실지로는 지정된 파일등록부가 없으므로 오유통보문을 보여준다.

이 식별이름에 한해서는 /etc/passwd파일에서 삭제하거나 주해(#)처리할것을 권고한다.

【관련명령문】

grpck : /etc/group파일검사명령문

pwconv

/usr/bin/pwconv

그림자암호의 사용

암호(password)파일을 그림자(shadow)정보로 변환하며 암호운영에서도 그림자방식을 사용한다.

형식 : pwconv

/etc/passwd파일은 일반 사용자도 읽기가 가능하므로 보안측면에서 약점을 가지고있다. pwconv는 /etc/passwd에 있는 자료를 변경하고 현재 존재하고있는 그림자정보에 포함시킨다. 암호가 설정되지 않은 사용자는 /etc/shadow에 추가시키지 않으며 암호기일이 지났거나 경고잠금 등의 정보를 가지고있는 /etc/login.defs에 정의된 값을 사용하게 된다.

최근의 배포판들에서는 기본으로 pwconv가 설정되여있어 특별히 설정할 필요가 없다.

해제를 위해서는 pwunconv명령을 사용한다.

이것은 그림자자료를 /etc/passwd형식으로 변경하고 /etc/shadow파일을 삭제하므로 사용하지 않는것이 좋다.

```
kcc@linux:~
# ls -al /etc/shadow
-r------- 1root root 1252 6월 6 23:49
/etc/shadow
```

/etc/shadow파일은 우의 권한과 같이 뿌리만이 읽을수 있으므로 침입공격의 위험을 보다 감소시킬수 있다.

【관련명령문】

grpconv : shadow집단을 만드는 명령문

pwd

/bin/pwd

현재경로보기

현재 사용하는 등록부의 절대경로를 보여주는 명령문이다.

형식 : pwd

pwd명령은 현재 작업중인 등록부의 절대경로를 보여준다.

실례로 /etc/httpd/conf등록부에로 이동한 후 작업중에 현재 등록부를 보려면 pwd명령으로 확인할수 있다.

```
kcc@linux:~
# cd /etc/httpd/conf/
# pwd
/etc/httpd/conf
```

어떤 작업등록부에서도 cd명령을 입력하면 $HOME등록부에로 이동한다.

```
kcc@linux:~
# cd
# pwd
/root
```

Q

quota /usr/bin/quota

디스크사용용량의 제한

특정한 사용자나 집단이 하드디스크를 많이 사용하여 다른 사용자들에게 피해를 주는것을 방지하기 위해 매 사용자나 집단에서 사용할수 있는 디스크 용량에 제한을 두는 명령문이다.

형식 : quota[설정값][사용자][집단]

-g : 식별이름과 관련된 집단의 할당몫(quota)정보를 보여준다.
-q : 할당몫제한을 초과한 사용자의 파일체식별이름보를 보여준다.
-u : 기정설정값이다. 사용자의 할당몫정보를 보여준다.
-v : 보다 자세한 할당몫정보를 보여준다.

quota는 사용자가 사용하는 디스크크기에 제한을 두어 체계관리의 효률성을 높일수 있게 한다. 기본적인 quota명령으로 체계접속한 식별이름의 할당몫설정내용을 볼수 있다.

```
kcc@linux:~
# quota
Disk quotas for user root (uld 09): none
```

아래와 같이 상세한 설정으로 할당몫(quota)을 설정해보자.

할당몫설정하기(RedHat 7.X이상)

1) /etc/fstab에서 할당몫제한을 두려는 파일체계에 usrqutoa를 입력한 후에 다시 기동한다.

```
LABEL=/ /ext2 defaults,usrquota 1 1
```

2) 우의 설정은 /에 제한을 두었으므로 /등록부에 touch명령으로 quota.user파일을 생성한다.

```
# touch /quota.user #chmod 600 /quota.user
```

3) convertquota명령으로 quota.user를 aquota.user자료파일로 생성한다.

```
# convertquota/
```

4) edquota admin명령으로 admin사용자의 쏘프트웨어(soft)와 하드웨어(hard)의 할당몫을 설정한다.

```
kcc@linux:~
# eduquota admin
dit block and inode quota for user admin: Device /dev/hda6 (/): Used
484KB, limits: soft=0 hard=0 Used 118 inodes, limits: soft=0 hard=0
```

5) repquota명령으로 파일체계의 할당몫을 간략형식으로 볼수 있다.

```
kcc@linux:~
# repquota -a
*** Report for user quotas on device /dev/hda6 (/)
Block grace time: 7 days; Inode grace time: 7 days
Block limits File limits
User used soft hard grace used soft hard grace
----------------------------------------------------------------------
root --- 42668 00 28 00
admin ---- 484 00 118 00
```

【관련명령문】

edquota : 할당량제한을 위해 매 설정을 편집하는 명령문

quotacheck : 할당량설정을 검사하는 명령문

quotaoff : 할당량제한을 해제하는 명령문

quotaon : 할당량제한을 설정하는 명령문

quotastats : 할당량상태를 보는 명령문

R

ramsize /usr/bin/ramsize

RAM디스크크기의 확인 및 설정

RAM디스크의 크기를 확인하고 설정하는 명령문이다.

형식 : ramsize[설정값][image[size[offset]]]

-o offset : 지정한 offset로 편위값을 지정한다.

ramsize명령은 RAM디스크의 크기를 보여주거나 설정할수 있다.

이 명령으로 핵심부환경변수(kernel image)를 지정하면 지정된 offset의 정보를 보여주게 된다.

```
kcc@linux:~

# rdev -r vmlinuz-2.4.13-1hl
Ramsize 0
```

ramsize명령은 rdev명령에 기호로 련결되여있다.

```
kcc@linux:~

# ls -al /usr/sbin//ramsize
lrwxrwxrwx  1 root   root    4 12월 14  2004 /usr/sbin/ramsize -> rdev
```

자세한 사항에 대해서는 rdev명령을 참고하기 바란다.

참 고

RAM디스크

콤퓨터의 체계 RAM령역의 일부를 사용하여 만들어진 가상적인 디스크장치이다. 특수한 프로그람이 이것을 마치도 디스크장치인것처럼 동작시킨다. RAM디스크는 조작체계로 하여금 디스크대신에 RAM으로부터 자료를 읽어들이고 RAM에 자료를 기록할수 있게 함으로서 접근시간을 훨씬 단축시키지만 전원이 꺼지면 자료가 지워진다.

ranlib

/usr/bin/ranlib

서고변환

문서고(archive)를 random서고로 변환한다.

형식 : ranlib 파일이름

ranlib명령은 지정한 문서고파일(archive file)명의 색인(index)을 생성하여 random서고로 변환한다. 이것은 -s명령과 같다.

참 고

문서고(archive)

문서고는 여러 파일을 한 덩어리로 뭉쳐놓은것으로서 흔히 예비보관이나 여러 파일을 한번에 이동시키기 위해 사용한다. 문서고는 단순히 파일들의 목록만을 포함할수도 있으며 하나의 파일등록부나 구조일람표의 아래에 파일들을 묶어 조직화할수도 있다. UNIX 및 Linux에서는 이러한 문서고를 만들고 해제하는데 흔히 tar편의프로그람을 사용한다. 다른 의미로는 웨브싸이트에 실린 내용중에서 열람할수 있는 이전의 내용, 혹은 웨브나 FTP에서 내리적재받을수 파일목록 등을 문서고라고 부르기도 한다.

rcp

/usr/bin/rcp

원격지파일복사

원격호스트의 파일을 복사하는 명령문이다.

형식 : rcp[설정값]파일1파일2
rcp[설정값]파일... 등록부

-p : 원본파일의 변경날자와 권한을 유지하여 복사한다.
-x : 호스트사이의 모든 정보전송을 DES암호화한다.
-r : 파일등록부안의 아래준위 파일등록부와 파일을 국부체계(local system)에로 복사한다.
D port : 지정한 포구로 접속을 한다.

rcp명령은 원격복사(remote copy)에 의해 망으로 련결되여있는 매 체계들간에 복사를 할수 있게 한다. r로 시작하는 명령들은 대부분 보안약점을 가지고있으므로 rcp대신에 scp명령을 사용할것을 권고한다.

[주의]
많은 체계들에서 보안상의 리유로 rcp대신에 rcp의 련결(link)을 사용하기도 한다.

참 고

DES(Data Encryption Standard): 자료암호화표준

매 64bit자료 블로크에 56bit길이의 암호열쇠를 적용한다. 이 과정은 여러가지 방식에서 실행될수 있으며 16회의 연산이 수행된다. DES가 강력한 암호화로 인정되고는 있지만 많은 회사들에서 세개의 열쇠가 련이어 적용되는 《triple DES》를 사용하고있다.

DES로 암호화된 통보문은 절대로 해독불가능한것은 아니다. 1997년초에 다른 암호화방식의 소유자인 RSA가 DES통보문의 해독에 10,000$의 상금을 걸었다.

인터네트에서 14,000명이상의 사용자들이 공동노력으로 다양한 암호열쇠들을 시험적용하여 결국 그 통보문을 해득하였는데 가능한 72000조개의 열쇠중에서 겨우 18000조개의 시험을 통해 그 열쇠가 발견되였다.

그러나 현재 DES암호화로 발송되는 통보문들가운데 이러한 종류의 코드해득노력의 영향을 받을것으로 보아지는것은 거의나 없다.

rdate

/usr/bin/rdate

시간맞추기

시간봉사기에서 시간정보를 얻어 체계의 시간을 변경한다.

형식 : rdate[설정값][호스트…]

-p : 호스트의 자료를 보여준다.
-s : 원격망호스트로부터 국부호스트시간을 설정한다.

rdate명령은 시간봉사기(time server)의 시간을 참고하여 체계의 시간을 변경한다. rdate -p설정값으로 time.kcc.net봉사기의 시간정보를 확인해보자.

```
kcc@linux:~
# rdate -p time.kcc.net
[time.kcc.net] Fn Jun 7 02:38:58 2004
```

[주의]

보통 명령행에서 입력하는 값은 체계를 다시 기동하면 모두 사라진다. 기동할 때마다 실행되여야 하는 명령들은 /etc/rc.d/rc.local파일에 추가하여야 한다. 우에 입력한 rdate -s time.kcc.net명령도 /etc/rc.d/rc.l ocal파일의 제일 아래행에 추가하여 체계를 기동할 때마다 시간봉사기에 시간을 맞추도록 할 필요가 있다.

rdate -s설정값으로 국부체계의 시간을 시간봉사기의 시간으로 변경시킬수 있다.

```
# rdate -s time.kcc.net
```

date명령으로 변경된 시간을 확인할수 있다.

```
kcc@linux:~
# date
2004. 06. 07. (금) 02:39:11 KST
```

rdev

/usr/bin/rdev

장치조사

root장치, swap장치, RAM디스크크기, video mode 등의 조사/설정을 위한 명령문이다.

형식 : rdev [설정값] [-o offset][image [value[offset]]]
rdev[-o offset][image[root_device[offset]]]

-r : rdev지령을 ramsize지령으로 사용한다.
-R : rdev지령을 rootflags지령으로 사용한다.
-v : rdev지령을 vidmode지령으로 사용한다.
-h : 도움말을 보여준다.

rdev는 RAM디스크의 크기, root flags, video mode 등의 상태를 보여주거나 설정할수 있게 한다.

체계기동에 사용되는 기동환경변수(boot image)안에는 root flags, video mode, RAM디스크용량 등의 정보가 들어 있다.

먼저 RAM 디스크의 용량을 확인해보자.

```
kcc@linux:~

# rdev -r /boot/vmlinuz
Ramsize 0
```

rdev -r명령에서 사용되는 RAM디스크의 크기값은 KB단위이다.

현재 RAM디스크로 사용되는 용량이 없다는것을 확인할수 있다.

rdev -r는 ramsize와 같은 명령으로서 ramsize는 rdev에 기호련결되여 있다. 다음으로 root flags를 확인해보자.

```
kcc@linux:~

# ls -al /usr/sbin/ramsize
lrwxrwxrwx    1 root   root  4 5월 19 16:17 /usr/sbin/ramsize -> rdev
```

rdev -R명령에서 사용되는 flags값은 뿌리파일체계를 mount할 때에 추가적인 정보를 담고있는 값이다. 일반적으로 flags값이 0이 아닌 경우 이 값은 핵심부가 뿌리파일체계를 읽기전용으로 마운트(mount)하도록 한다.

rdev -R는 rootflags와 같은 명령으로서 rootflags는 rdev에 기호적으로 련결되여있다.

```
kcc@linux:~
# ls -al /usr/sbin/rootflags
lrwxrwrwx 1 root  root  4 5월 19 16:17 /usr/sbin/rootflags -> rdev
```

마지막으로 video mode를 확인해 보자

```
kcc@linux:~
# rdev -v /boot/vmlinuz
Video mode 65535
```

vidmode명령에서 사용되는 mode값은 다음과 같은 video mode를 지정한다.

```
-3=Prompt
-2=Extended VGA
-1= Normal VGA
0=as if "0" was pressed at the prompt
1=as if "1" was pressed at the prompt
2=as if "2" was pressed at the prompt
n=as if "n" was pressed at the prompt
```

value값이 지정되지 않으면 image값은 다만 현재의 설정을 확인하기 위해 사용된다. rdev -v는 vidmode와 같은 명령으로서 vidmode는 rdev에 기호련결되여있다.

```
kcc@linux:~
# ls -al /usr/sbin/vidmode
lrwxrwxrwx  1 root  root  4 5월 19 16:17/usr/sbin/vidmode -> rdev
```

참 고

Apache, PHP, MYSQL의 설치

개개의 제공싸이트로부터 최신원천을 내리적재받는다.

그리고 이 원천을 /usr/local등록부로 가져와 압축을 푼다. 설치는 보통 다음의 순서로 쉽게 할수 있다. 개개의 ./configure부분에 추가한 설정에 대해서도 기본적으로 사용하는것을 아래에 주었다.

```
kcc@linux:~

# tar xvzf apache_판본.tar.gz: mv apache_판본 apache
# tar xvzf mysql-판본.tar.gz: mv php-판본 php
# tar xvzf php-판본.tar.gz: mv mysql-판본 mysql
# cd /usr/local/mysql        MySQL 설치
# ./configure —prefix=/usr/local/mysql \
>—localstatedir=/usr/local/mysql --with-charset=euc_kp
# make;make install;
# /cd/usr/local/mysql
# ./bin/mysql_install_db
# ./bin/safe_mysqld &
# cd /usr/local/apache       Apache 경로 설정
# ./configure —prefix=/usr/local/apache
# cd /usr/local/php           PHP설치
# ./configure —with-apache=./apache \
>—with-mysql=/usr/local/mysql  enable-track-vars=yes
# make:make install
# cd /usr/local/apache/src    Apache 설치
# ./configure
# cd ..
# ./configure —prefix=usr/local/apache \
>—activate-module=src/modules/php4/libphp4.a
# make:make install
# cd /usr/local/apache/bin
# ./apachectl start
```

readonly

/usr/bin/readonly

읽기전용

파일을 읽기전용으로 표시하는 명령문이다.

형식 : readonly [설정값][이름…]

-f : 이름에 해당하는 함수를 읽기전용으로 표기한다.
-p : 모든 읽기전용변수목록을 보여준다.

readonly명령은 주어진 이름들을 읽기전용으로 표기하고 다음에 나올 대입문에 의해 이름들의 값이 바꾸어지지 않도록 해준다. --인수는 나머지의 인수에 대한 검사를 하지 않도록 지시한다. 잘못된 설정값이 있거나 이름들 중의 하나라도 적합한 쉘 변수이름이 아닌 경우 또는 -f 다음에 나온 이름이 함수가 아닌 경우를 제외하고는 반환상태값이 0이다.

자세한 사용법에 대해서는 쉘 스크립트부분을 참고하면 된다.

reboot

/usr/bin/reboot

체계재기동

체계를 완료하고 다시 기동시키는 체계관리명령문이다.

형식 : reboot [설정값]

-d : wtmp파일에 기록을 남기지 않는다.
-f : 강제로 재기동한다.
-n : 재기동할 때에 sync하지 않는다.
-w : 실지로 재기동하지 않고 /var/log/wtmp기록을 남긴다.

reboot명령은 다른 사용자가 체계를 사용하지 않을 때 체계를 재기동하는 명령문이다. 이것은 shutdown -r now명령이나 init 6명령과 같다.

shutdown명령은 보통 체계가 수행중이고 사용자가 현재 작업중의것을 보관할 시간적여유를 주는 경우 통보문에 의해 이것을 알리게 된다. 그러나 reboot는 경고없이 체계관리자가 즉시 재기동하게 된다.

reboot명령은 syslog명령으로 /var/log/message와 체계접속사용의 통계리력인 /var/log/wtmp에 이것을 보관하게 된다.

renice

/usr/bin/renice

우선권순위변경

실행중인 프로세스의 작업우선권순위를 변경하는 명령문이다.

형식 : renice 우선권[설정값][프로세스]

설정값
+num : 현재 프로세스의 우선권을 지정해준 num수만큼 낮춘다.
-num : 현재 프로세스의 우선권을 지정해준 num수만큼 높여준다.
-g : 집단ID를 지정
-u : 사용자ID를 지정
-p : 프로세스 ID 를 지정(초기값)

renice명령은 실행중인 하나 또는 그 이상의 프로세스의 우선권을 변경시킨다. 이것을 nice명령과 비교해보자.

renice의 매 설정값들에서 사용되는 첨수는 프로세스ID와 프로세스집단ID, 사용자ID 등이다. 프로세스집단의 우선권변경은 그에 해당되는 집단의 모든 프로세스들에 적용된다. 사용자의 우선권변경도 그에 해당되는 사용자의 모든 프로세스들에 적용된다. 초기값으로 우선권변경에 영향을 받는 프로세스는 지정한 프로세스뿐이다.

실례는 nice명령에 함께 설명되어있다.

reset

/usr/bin/reset

말단을 초기화하는 명령문

말단을 초기화하는 명령문이다.

형식 : reset [설정값][말단]

- : 현재 말단의 종류를 보여준다.
-e ch : 지정된 ch문자를 말단에서 ERASE문자로 설정한다.
-i ch : -e설정값과 류사하지만 지정된 ch 문자를 말단에서 새치기문자로 설정한다.
-k ch : 지정한 ch문자를 말단에서 LINE KILL문자로 설정한다.
-m mapping : 지정한 mapping을 말단에서의 포구형태로 설정한다.

clear는 현재 말단에 보이는것을 화면상에서 완전히 지운후 유표를 화면의 맨 웃단으로 이동시키는 명령이지만 reset는 말단을 초기화한다.

reset명령은 프로그람이 완료되여 말단이 예상치 못한 상태로 일반적인 명령문이 인식되지 않는 경우에 Ctrl+j clear Ctrl+j 명령으로 말단을 초기화할수도 있다.

reset명령은 tset명령의 기호련결파일이다.

```
kcc@linux:~
# ls -al /usr/bin/reset
lrwxrwxrwx  1 root    root   4 12월 14  2004 /usr/bin/reset -> tset
```

【관련명령문】

clear : 화면을 완전히 지우고 유표를 화면의 제일 웃단으로 이동시킨다.

restore

/sbin/restore

dump예비보관파일을 복구하는 명령문

dump예비보관파일을 복구한다.

형식 : restore[설정값]

restore -C [-ckMvy][-b blocksize][-D filesystem][-f file][-s fileno] [-T directory]

restore -I [-chkmMNuvy][-b bloksize][-f file][-Q file][-s fileno [-T directory]

restore -R[-ckMNuvy] [-b blocksize][-f file] [-sfileno][-T directory]

restore -r [-ckMNuvy][-b blocksize][-f fileno][-T directory]

restore -t [-chkMNuvy][-b blocksize][-f file][-Q file][-s fileno] [-T directory][-X filelist] [파일이름]

restore -x [-chkmMNuvy][-b blocksize][-f file][-Q file][-s fileno] [-T directory][-X filelist] [파일이름]

dump명령문은 복잡한 파일체계의 예비보관편의프로그람으로서 일반적으로 규모가 큰 설치와 망에서 쓰인다. restore명령문은 이 dump로 예비보관된 파일을 복구한다.

rev

/usr/bin/rev

파일내용을 거꾸로 바꾸기

파일의 내용을 행단위로 읽고 자모문자순서와 반대로 바꾸어 보여주는 명령문이다.

형식 : rev[파일]

rev명령은 지정한 파일을 행단위로 읽어 그 행의 내용을 역변환해서 보여준다. 파일이 지정되지 않으면 표준입력으로 입력받는다.

genie라는 파일을 한번 보자.

```
kcc@linux:~
# cat genie
[Desktop Entry]
Comment[ko]=
Exec=kdesu /usr/bin/genie
Icon=/usr/share/pixmaps/kcccom/update-hl-32.png
Mime Type=
Name=genie
```

rev명령은 genie라는 파일의 매 행의 내용을 거꾸로 보여준다.

```
kcc@linux:~
# rev genie
]yrtnE potkseD[
=Jok[tnemmoC
eineg/nib/rsu/ usedk=cexE
gnp.23-lh-etadpu/moccck/spamxip/erahs/rsu/=nocI
=epyTemiM
eineg=emaN
```

rexec

/usr/bin/rexec

원격실행

원격호스트에 접속하여 명령을 실행한다.

형식 : rexec[설정값] -l username -p password host command

설정값
-l username : 원격호스트에 접속할 사용자명을 지정한다.
-p password : 원격호스트에 접속할 사용자명의 암호를 지정한다.
-n : 접속할 사용자와 암호입력독촉(password prompt)을 보여준다.
-h : 사용법을 보여준다.

rexec명령은 지정한 원격호스트에 실행할 명령을 입력하여 국부체계에서 실행한 명령의 결과를 볼수 있게 하는 명령문이다. r로 시작하는 대부분의 명령문들은 보안상의 약점을 가지고있으므로 rexec는 될수록 사용하지 않는 것이 좋다. 그러나 rexec명령에 대해 알아둘 필요는 있으므로 아래에서 간단히 언급한다.

rexec.linuxroot.co.kp호스트의 date명령을 실행하여 이 호스트의 시간을 한번 확인해보자.

```
# rexec rexec.linuxroot.co.kp.date
```

이 실례에서 국부체계는 $HOME/.netrc파일에 원격호스트의 사용자와 암호를 보관하게 된다.

아래의 실례는 rexec.linux.root.co.kp호스트에서 admin식별이름의 home등록부목록을 국부체계에 보여주게 된다.

```
# rexec rexec.linuxroot.co.kp ls -l /home/admin
```

[주의]

rexec를 사용하기 위해서는 원격호스트가 rexec 봉사를 지원하여야 한다. rexec 봉사는 super데몬인 xinitd에 포함되여있으며 /etc/xinitd.d/rexec파일을 변경하여 사용할수 있다.

rlogin

/usr/bin/rlogin

원격호스트접속

자주 접속하는 호스트를 허용하여 자동적으로 telnet접속을 할수 있는 명령문이다.

형식 : rlogin 호스트 [설정값]

설정값

-t ttytype : tty형태를 지정한다.

-l username : 접속할 사용자식별이름을 지정한다.

rlogin은 telnet와 류사한 기능을 한다. 그러나 rlogin은 허용한 호스트간에 자동적으로 체계접속할수 있다. 만일 linux1.linuxroot.co.kp에서 linux2.linuxroot.co.kp로 telnet접속을 하는 경우, 매번 식별이름을 입력한후에 암호를 입력해야 한다.

그러나 rlogin을 사용하면 linux1.linuxroot.co.kp호스트를 rlogin으로 허용하여 자동적으로 체계접속하도록 할수 있다. r로 시작하는 대부분의 명령문들이 보안상의 취약점을 로출시키는데 rlogin명령도 례외로 되지 않는다. 될수록이면 rlogin명령대신에 ssh명령을 사용할것을 권고한다.

RedHat 7.X판본은 /etc/xinetd.d/rlogin파일에서 기본적으로 봉사가 금지되여있다.

```
kcc@linux:~

# cat /etc/xinetd.d/rlogin
# default: on
# description: rlogind is the server for the rlogin(1) program. The
server\
#       provides a remote login facility with authentication based
an \
#       privileged port numbers from trusted hosts.
service login
```

```
{
disable= yes
        socket_type        =stream
        wait               = no
       user                = rooot
        log_on success   +=USERID
        log_on failure  +=USERID
        server           =/usr/sbin/in.rlogind
}
```

【관련명령문】

telnet : 원격 telnet봉사기에 접속하기 위한 의뢰기
ssh : 원격 SSH봉사기에 접속하기 위한 의뢰기

rm /bin/rm

파일삭제

파일과 등록부를 삭제하는 명령문이다.

형식 : rm[설정값]파일이름

-d, —directory : rmdir지령대신에 unlink'와 함께 경로를 지운다.

-f, —force : 삭제할 파일이 없는 경우 아무 통보문도 보여주지 않고 그냥 넘어간다.

-i —interactive : 삭제하기전에 사용자에게 삭제여부를 확인한다.

-r -R —recursive : 하위등록부를 포함한 모든 내용을 삭제한다.

-v —verbose : 삭제되는 파일의 정보를 화면에 출력한다.

-R : -r설정값과 같은 역할을 한다.

—help : 표준출력으로 도움말을 보여준다.

—version : 표준출력으로 판본정보를 보여준다.

rm명령은 파일등록부나 파일을 삭제하는 명령문이다.

보통 배포판에서는 $HOM/.bashrc파일에 alias rm=' rm -i' 를 설정하여 실수로 파일이 삭제되는것을 피할수 있게 한다. -i설정값은 삭제하기전에 사용자로 하여금 삭제여부를 확인하게 된다.

```
kcc@linux:~
# rm httpd
rm: remove tv? y
```

만일 확실하게 삭제할 파일이나 파일등록부인 경우에는 -rf설정값으로 등록부를 포함한 아래준위 파일등록부와 파일의 모든 삭제를 할수 있다.

Netscape열람기를 설치하기 위해 림시등록부로 생성된 netscape-installer가 있다.

```
kcc@linux:~

# ls -al netscape-installer/
합계 308
drwxr-xr-x  2 root root 4096 4월 25 09:43 .
drwxr-x—  42 root root 4096 6월 7 14:21 ..
-rw-r—r--  1 8482 wheel 7290 3월 15 17:15 README
-rw-r—r--  1 8482 wheel 12496 3월 15 17:17 config.ini
-rw-r—r--  1 8482 wheel 2807 3월 15 17:15 installer.ini
-rw-r  r--  1 8482 wheel 17695 3월 15 17:15 license.txt
-rwxr-xr-x 1 8482 wheel 1524 3월 15 17:15 netscape-installer
-rwxr-xr-x  1 8482 wheel 247626 3월 15 17:15 netscape-installer-bin
```

```
# rm -rf netscape-installer
```

[주의]

이것은 rm -rf등록부로 한꺼번에 삭제할수 있는것이다.

—설정값 다음에 오는것은 설정값이 아닌것으로 인식한다. 즉 파일이름이 -f라는 파일을 삭제하려고 하는 경우에 다음의 두가지 방법을 사용한다.

#> rm — -f 혹은 #> rm ./-f

rm -ir로 yes/no를 선택하여 아래준위등록부내의 목적하는 파일만을 삭제할수도 있다.

```
kcc@linux:~

# rm -ir Xfree86-4.2/
rm: descend into derecory xfree86-4.2/' ? y
rm: remove xfree86-4.2//Xfree86-100dpi-fonts-4.2.0-8.i386.rpm' ? n
rm: remove xfree86-4.2//Xfree86-4.2.0-8.i386.rpm' ? n
rm: remove xfree86-4.2//Xfree86-75dpi-fonts-4.2.0-8.i386.rpm' ? n
rm: remove xfree86-4.2//Xfree86-ISO8859-15-100dpi-fonts-4.2.0-8.i386.rpm' ? n
rm: remove xfree86-4.2//Xfree86-ISO8859-15-75dpi-fonts-4.2.0-8.i386.rpm' ? n
rm: remove xfree86-4.2//Xfree86-ISO8859-2-100dpi-fonts-4.2.0-8.i386.rpm' ? n
rm: remove xfree86-4.2//Xfree86-ISO8859-2-75dpi-fonts-4.2.0-8.i386.rpm' ?
```

【관련명령문】

rmdir : 파일등록부 삭제명령문

rmail

/usr/bin/rmail

원격우편보기

uucp를 통해 받은 원격우편을 다루는 명령문이다.

형식 : rmail[설정값]사용자

-D domain : 기본값인 UUCP대신에 지정한 domain을 설정한다.
-T : 오유수정방식

UUCP환경에서 수신주소와 우편통보문을 전송하려고 하는 린접호스트상에서 rmail명령을 주면 우편을 전송하게 된다. rmail명령은 통보문을 다른 호스트에로 전송하여 그 통보문이 목적호스트에 도착하게 한다.

이 명령은 현재 거의나 쓰이지 않는다.

참 고

UUCP (UNIX-to-UNIX Copy Protocol)

UUCP는 다른 UNIX체계들간에 파일을 복사하거나 다른 체계상에서 실행될 명령문들을 전송하기 위한 UNIX프로그람집합이다. UUCP의 주요한 명령문들은 다음과 같다.

uucp : 특정파일을 지정한 다른 체계에로 복사할것을 요청

uux: UNIX명령문을 실행될 다른 체계에로 전송

uucico : 복사를 진행하고 전송된 명령문수행을 시작하는 프로그람으로서 UNIX체계상에서 실행된다. 보통 이 프로그람은 하루에도 수회에 걸쳐 실행되는데 그 동안에 복사(uucp) 및 명령문(uux)요청들은 uucico프로그람이 실행될 때까지 대기렬(que)에 보관된다.

uuxqt : uucico프로그람에 의해 시작된후에 uux에 의해 전송된 명령문들을 수행한다.

rmdir

/bin/rmdir

등록부삭제

빈 등록부를 삭제하는 명령문이다.

형식 : rmdir[설정값]등록부

-p, —parents : 웃준위경로도 지운다. 이 명령은 해당한 웃준위경로의 내부가 비여있어야 가능하다.
—verbose : 진행상황을 자세하게 보여준다.
—help : 도움말을 보여주고 완료한다.
—version : 판본정보를 보여주고 완료한다.

rmdir는 아래준위 파일등록부나 파일이 없는 등록부를 지우는 명령문이다. 만일 아래준위 파일등록부나 파일이 들어있는 등록부를 삭제하려면 rm -r명령으로 한다. rmdir명령은 등록부가 비여있지 않으면 삭제되지 않는다.

비여있지 않는 maililst 파일등록부를 한번 삭제해보자.

```
kcc@linux:~
# rmdir maillist/
rmdir: maillist/' : 등록부가 비여있지 않음
```

이때에는 먼저 아래준위등록부의 파일이나 등록부를 rm명령으로 삭제해야 한다.

```
kcc@linux:~
# rm -rf maillist/*
# rmdir maillist/
```

【관련명령문】

rm : 파일이나 등록부를 삭제하는 명령문

rmmod

/sbin/rmmod

모듈제거

사용할수 있는 상태의 핵심부모듈을 주기억기에서 제거하는 명령문이다.

형식 : rmmod[설정값]모듈명

-a, —all : modprode -a로 자동적재된 모듈까지 삭제한다.
-h —help : 사용법을 보여준다.
-r —stacks : 모듈stack를 제거한다.
-s —syslog : syslog에 통보문을 남긴다.
-v —verbose : 진행상태를 상세히 보여준다.
-v —version : 모듈의 판본을 보여준다.

rmmod명령은 체계에 적재된 모듈을 제거할 때에 사용하는 명령문이다.

이 명령문은 관련명령문들과 매우 밀접한 련관이 있으므로 해당한 명령문들을 한번씩 련관시켜보기를 바란다.

현재 체계에 적재되어있는 모듈들을 lsmod명령으로 확인할수 있다.

```
kcc@linux:~

# lsmod
Module        Size    Used by     Tainted:P
ip_tables     11200   0 (unused)
smbfs         35552   2 (autoclean)
nls_cp949     141440  1 (autoclean)
ide-cd        27360   1 (autoclean)
cdrom         28480   0 (autoclean) [ide-cd]
ymfpci        42660   0 (autoclean)
uart401       6560    0 (autoclean) [ymfpci]
sound         59052   0 (autoclean) [uart401]
ac97_codec    9504    0 (autoclean) [ymfpci]
soundcore     4324    4 (autoclean) [ymfpci sound]
autofs        10948   0 (autoclean) (unused)
8139too       17440   2
ipx           16404   0  (autoclean)
ext3          61600   2  (autoclean)
ibd           40452   2  (autoclean)[ext3]
usb-uhci      21764   0  (unused)
usbcore       51744   1  [usb-uhci]
```

그중 ip_tables모듈을 한번 삭제해보자.
아무러한 오유가 없는것으로 보아 모듈이 제거되였다고 볼수 있다.

```
#  rmmod ip_tables
```

【관련명령문】

depmod : 모듈적재를 위한 의존성관계를 다루는 명령문
modprobe : insmod보다 우수한 모듈적재명령문
insmod : 모듈적재명령문

rootflags

/sbin/rmmod

뿌리장치설정명령문

뿌리파일체계를 마운트(mount)한다.

rev -R설정값의 명령으로서 rootflags는 기호련결파일이다.
자세한 사항에 대해서는 rdev명령을 참고하기 바란다.

```
kcc@linux:~
# ls -al /usr/sbin/rootflags
lrwxrwxrwx  1 root  root   4 12월 14 2004 /usr/sbin/rootflags-> rdev
```

route

/sbin/route

경로표보기

IP경로표를 보여주거나 다루는 명령문이다.

형식 : route[설정값][명령문]

-A family : 지정한 주소를 설명하여준다.
-v,—verbose : 보다 상세한 정보를 보여준다.
-n,—numeric : Destination항목값을 모두 IP주소로 보여준다.
-e,—extend : 보다 많은 경로표정보를 보여준다.
-F,—fib : 기본적인 경로정보를 보여준다.
-C,—cache : routing cache정보를 보여준다.
-h, —help : 도움말을 보여준다.
-V,-version : 판본정보를 보여준다.
add [-netl-host] IP주소 [gw gateway][netmask 네트마스크] [mss tcp-mss][dev장치]
del IP 주소

route명령을 사용하면 망경로표에 수동항목을 작성할수 있다.
이것은 /etc/hosts에 설정한 호스트 alias설정이나 /etc/resolve.conf파일에 설정된 이름봉사기(name server)의 정보에서 주소를 해석하게 된다.

```
kcc@linux:~

# route
Kernel IP routing table
Destination  Gateway     Genmask     Flags Metric Ref  Use
Iface
192.168.0.0  *        255.255.255.0   U   0   0      0 eth1
61.40.233.0  *        255.255.255.0   U   0   0      0 eth0
127.0.0.0    *        255.0.0.0       U   0   0      0 lo
default      61.40.233.1  0.0.0.0.    UG  0   0      0 eth0
```

여기에서 eth1망장치는 192.168.0.0의 망주소를 가지며 Genmask 즉, 네트마스크는 255.255.255.0의 값을 가진다.

그리고 eth0망장치는 61.40.233.0의 망주소를 가지며 255.255,255.0의 네트마스크값을 가진다. lo장치는 망기판이 인식되여있을 때 자동적으로 인식하는 장치로서 자기자신의 IP 즉, 국부호스트주소를 가지고있는 127.0.0.1에 대한 장치경로표이다. 경로표안에 있는 Destination항목의 기정값은 게트웨이주소를 말하는것으로서 61.40.233.1의 값을 가지고있다. -n 설정값은 호스트명대신에 4분표기법으로 된 주소를 나타낸다.

```
kcc@linux:~

# route -n
Kemel IP routing table
Destination   Gateway      Genmask        Flags Metric Ref  Use Iface
192.168.0.0   0.0.0.0      255.255.255.0  U     0      0      0 eth1
61.40.233.0   0.0.0.0      255.255.255.0  U     0      0      0 eth0
127.0.0.0     0.0.0.0      255.0.0.0      U     0      0      0 lo
0.0.0.0       61.40.233.1  0.0.0.0        UG    0      0      0 eth0
```

경로표에 나타난 기본게트웨이것을 한번 삭제해보자. 이때에는 route del 명령과 함께 Destionation의 값인 기정값을 지정한다.

이와 함께 게트웨이것을 삭제하므로 gw나 gateway를 입력하며 게트웨이 주소를 지정하면 된다.

```
# route del default gw 61.40.233.1
```

기본 게트웨이의 추가도 꼭같은 방식이나 route del명령을 route add명령으로만 입력한다.

```
# route add default gw 61.40.233.1
```

참　고

경로기(Router)

경로기는 같은 전송규약을 사용하는 분리된 망을 련결하는 장치로서 망계층간을 서로 련결한다. 경로기는 망다리(Bridge)가 가지는 기능에 추가하여 경로표에 따라 다른 망 또는 자체망안의 마디(node)를 결정한다. 그리고 여러 경로가운데서 가장 효률적인 경로를 선택하여 파케트를 보낸다. 경로기는 흐름조종을 하며 인터네트망의 내부에서 여러 부분망을 구성하고 다양한 망관리기능을 수행한다. 망다리와 경로기와의 차이점을 간단히 보면 경로기는 망계층까지의 기능을 담당하고있으면서 경로설정을 해주는 반면에 망다리는 자료련결층까지의 기능만으로 목적지주소에 따른 선별 및 간단한 경로결정을 한다.

게트웨이(Gateway)

게트웨이는 다른 망에로 들어가는 입구역할을 하는 망점이다. 경로조종의 관점에서 보면 인터네트는 많은 게트웨이마디와 호스트마디로 구성된 망이라고 할수 있다. 망사용자들의 콤퓨터들과 웨브페지와 같은 내용물을 제공하는 콤퓨터들이 호스트마디에 속하며 한 기업소의 망내에서 통신량(traffic)을 통제하는 콤퓨터들이나 인터네트봉사 제공자들의 콤퓨터가 바로 게트웨이마디에 속한다.

rpm

/bin/rpm

RedHat패키지관리자

프로그람을 체계에 쉽게 설치하기 위한 명령문으로서 RedHat회사에 의해 만들어졌지만 대부분의 배포판들에서 사용할수 있는 패키지관리자이다.

형식 : rpm[설정값]

-w : 오유수정에 관한 정보를 상세히 출력한다.

-keep-temps : 림시파일을 삭제하지 않는다(/tmp/rpm-*). rpm을 오유수정할 때에만 주로 사용한다.

-quiet : 최소한의 출력만을 하며 다만 오유통보문만을 출력한다.

-help : 보다 상세한 사용법을 보여준다.

-version : 사용중인 rpm의 판본을 한행으로 표시한다.

-rcfile <파일> : /efc/rpmrc 또는 $HOM/.rpmrc를 사용하지 않고 <file>을 사용하도록 한다.

-root <dir> : 모든 동작에 대하여 최상위등록부를 주어진 파일등록부로 설정하고 작업한다. 례하면 설치할 때 주어진 파일등록부를 /라고 생각하고 그것을 기준으로 설치해나간다.

설치방식	rpm -i[설치설정값]<패키지파일>+
문의방식	rpm -q[문의설정값]
검증방식	rpm -v l -y l —verify[검증설정값]
서명확인방식	rpm—checksig<패키지파일>+
제거방식	rpm -e<패키지이름>
제작방식	rpm -bo[제작설정값]<패키지spec>+

많이 사용하는 설정값에 대해 우에서 설명하였는데 아래에서는 매 설정값들에 대해 보다 상세한 설명을 주었다.

설치설정값

rpm설치명령의 일반적인 형태는 다음과 같다.

```
rpm -i[설치설정값들]<패키지파일>+
```

• Force : --replacepkgs, --replacefiles, --oldpackage 등을 모두 사용한것과 같다.

• h, --hash : 패키쥐를 풀 때 #를 표시한다. 총개수는 50개이다. 보다 좋은 출력을 위해서는 -v를 함께 사용한다.

• Oldpackage : 새로운 패키쥐를 지우고 이전의 패키쥐로 교체할 때에 사용한다.

• Percent : 패키쥐파일을 풀 때에 백분률(%)표시를 한다.

• Replacefiles : 이미 설치된 다른 패키쥐의 파일우에 덧쓰기하면서라도 패키쥐를 강제로 설치한다.

• Replacepkgs : 패키쥐가 이미 설치되여있는 경우에도 다시 설치한다.

• root<등록부> : <등록부>를 root로 하는 체계에 설치를 수행한다. 자료기지는 <등록부>아래에 갱신되며 pre 또는 post스크립트는 <등록부>로 chroot() 한 후, 실행됨을 의미한다.

• Noscripts : preinstall, postinstall스크립트를 실행하지 않는다.

• Excludedocs : 문서라고 표시되여있는 파일(man페지와 texinfo문서)은 설치하지 않는다.

• includedocs : 문서파일을 포함한다. 이 설정값은 rpmrc파일에 excludedocs : 1이 명시되여있을 때에만 필요하다.

• test : 패키쥐를 실지로 설치하지는 않고 충돌사항이 있는가를 검사하여 보고한다.

• -U, --upgrade : 현재 설치되여있는 패키쥐를 새로운 판본의 RPM으로 갱신한다. -i 설정값과 같은데 종전의 판본이 자동적으로 지워지는것이 다른점이다.

문의설정값

rpm문의설정값의 일반적인 형식은 다음과 같다.

```
rpm -q[문의설정값]
```

여기에서는 패키쥐정보가 표시될 형식을 결정해주어야 한다.

--queryformat 설정값뒤에 형식문자렬을 써주면 된다.

문의형식은 표준 prinif형식을 약간 변경한것이다. 형식은 정적문자렬(개행문자, tab, 그리고 다른 특수문자에 대한 표준 C문자 escape표기)과 prinrf형식지정자로 구성되여있다. rpm은 이미 출력형태를 알고있으므로 형식지정자는 생략하고 {}로 묶어서 머리부꼬리표(header tag)의 이름으로 바꾸어 주어야 한다. 꼬리표명에서 RPMTAG_부분은 생략해야 하며 꼬리표명앞에는 -를 기입해주어야 한다.

례를 들어 문의대상패키쥐의 이름만 출력하려고 하는 경우에 %{NAME}을 형식문자렬로 사용해야 한다. 패키쥐명과 배포판에 관한 정보를 두개의 column으로 표시하려고 할 때는 %-30{NAME}%{DISTRIBUTION}라고 입력하여 준다.

rpm은 --querytags설정값을 주면 인식하고있는 모든 꼬리표의 목록을 보여준다.

문의설정값에는 패키쥐선택과 정보선택의 2가지가 있다. 패키쥐선택설정값에 대한 상세설명은 다음과 같다.

• **<패키쥐이름>** : <package_name>라는 이름의 패키쥐에 대한 문의를 수행한다.

• a : 모든 패키쥐에 대하여 문의를 수행한다.

• whatrequites**<기능>** : 제대로 동작하기 위해 <기능>을 필요로 하는 모든 패키쥐에 대하여 문의를 수행한다.

• whatprovides**<가상>** : <virtual>기능을 제공하는 모든 패키쥐에 대하여 문의를 수행한다.

• f**<파일>** : <파일>을 포함하는 패키쥐에 대하여 문의를 수행한다.

• F : -f와 같은데 파일이름을 표준입력으로 읽는다.

• p**<패키쥐파일>** : 설치된 또는 설치되지 않은 <패키쥐파일>에 대하여 문의를 수행한다.

• P : -p와 같은데 패키쥐파일이름을 표준입력으로 읽는다.

한편 정보선택설정값에 대한 상세설명은 다음과 같다.

• -I : 패키쥐이름, 판본, 설명 등의 정보를 출력한다. 만일 queryformat가 주어져 있으면 그것을 리용하여 출력한다.

• R : 현재 패키쥐가 의존하고있는 패키쥐목록을 보여준다(--requires와 같음).

• provides : 패키쥐가 제공하는 기능을 보여준다.

• l : 패키쥐안의 파일을 보여준다.

• s : 패키쥐안에 포함된 파일의 상태를 보여준다(-l은 포함). 매 파일의 상태는 normal(정상), not installed(설치되지 않음), replaced(교체) 등의 값을 가진다.

• d : 문서파일만을 보여준다(-l은 포함).

• c : 설정파일만을 보여준다(-l은 포함).

• scripts: 설치, 제거과정에 사용되는 쉘스크립트가 있는 경우, 그 내용을 출력한다.

• dump: 다음과 같은 파일정보를 dump한다 : 경로, 크기, 수정일, MD5검사합, 권한, 소유자, 집단, 설정파일의 여부, 문서파일의 여부, rdev, 기호련결의 여부 등 최소한 -l,-c,-d 등의 설정값들중의 하나가 사용되여야 한다.

검증설정값

rpm검증설정값의 일반적인 형태는 다음과 같다.

```
rpm -Vlyl  verify [검증설정값]
```

설치되여있는 파일들에 대하여 rpm자료기지에 보관된 내용과 원본패키쥐(original package)의 내용을 비교한다.

검증내용은 크기, MD5검사합, 권한, 형식, 소유자, 집단 등이며 차이점이 발견되면 그에 대해 출력한다. 패키쥐지시설정값은 패키쥐문의설정값에서와 같다.

출력형식은 8문자의 문자렬이다. 《c》는 설정파일을 의미하며 파일이름을 나타낸다. 개개의 8개문자는 RPM자료기지에 보관된 속성과 비교한 결과를

나타낸다. 점(.)은 이상없음을 나타낸다. 비교결과 문제점이 발견된 경우에 다음과 같은 문자가 나타난다.

5	MD5검사합
S	파일크기
L	기호련결
T	갱신날자
D	장치
U	사용자
G	집단
M	권한과 파일형식을 포함한 방식

서명확인

rpm서명확인명령은 다음과 같다.

```
rpm   checksig <패키쥐파일>+
```

패키쥐의 원본여부를 가려내기 위하여 패키쥐안에 포함된 PGP서명을 검사한다. 설정정보는 /etc/rpmrc에서 읽어낸다.

제거설정값

rpm제거명령의 일반적인 형태는 다음과 같다.

```
rpm -e <패키쥐파일>+
```

- noscripts : preunistall, postuninstall스크립트를 실행하지 않는다.
- nodeps : 패키쥐를 제거할 때에 의존성을 검사하지 않는다.
- test : 실지로 패키쥐를 제거하는것은 아니고 검사하여본다.

제작설정값

rpm제작명령의 일반적인 형식은 다음과 같다.

```
rpm -bo [제작설정값]<패키쥐스페크>+
```

-bo 는 제작단계와 제작할 패키쥐를 나타내는것으로서 아래의것들가운데서 하나의 값을 가진다.

• bp : 스페크파일의 《%prep》단계를 실행한다. 보통 원천을 풀고 림시수정기(Patch)를 가하는 작업이다.

• bl :《목록검사》를 한다. 《%files》단락은 확장마크로이다. 이 파일들의 존재여부를 알아본다.

• bc :《%build》단계를 수행한다(prep단계를 수행한 후). 보통 make에 해당하는 작업을 수행한다.

• bi :《%instalr》단계를 수행한다(prep, build 단계를 거친 후). 보통 make install에 해당하는 작업을 한다.

• bb : 2진패키쥐를 만든다(prep, build, istall단계를 수행한 후).

• ba : 2진형식과 원천형식의 패키쥐를 만든다(prep, build, install 단계를 수행한 후).

다음의 설정값도 사용할수 있다.

• short-circuit : 중간단계를 거치지 않고 지정한 단계에로 직접 이동한다. -bc, -bi와 함께만 사용할수 있다.

• timecheck :《시간검사》를 0(불가능)으로 설정한다. 이 값은 rpmrc에서 《timecheck:》로 설정할수 있다. 시간검사값은 초로 표시되는데 파일이 packaging되는 최대시간을 정한다. 시간을 초과하는 파일들에 대하여 경고통보문이 출력된다.

• clean : 패키쥐를 만든 후 build등록부를 지운다.

• test : 어떠한 build단계도 거치지 않는다. 스페크파일을 검사할 때에 유용하다.

• sign : 패키쥐안에 PGP 서명을 넣는다. 패키쥐를 누가 만들었는가를 확인할수 있다.

• **rpm--rfecompile <원천패키쥐파일>+ 또는 rpm --rebuild <원천패키쥐파일>+** : rpm은 주어진 원천패키쥐를 설치하고 prep번역과 설치를 해준다. --rebuild는 새로운 2진패키쥐도 만들어 준다. 패키쥐만들기를 마치면

build등록부는 --clean설정값에서와 마찬가지로 지워진다. 패키쥐로부터 나온 원천과 스페크파일은 삭제된다.

이미 있던 RPM에 서명하기

```
rpm -resign <2진패키쥐파일>+
```

패키쥐파일에 새로운 서명을 한다. 기존의 서명은 삭제된다.

서명기능을 사용하기 위해서는 PGP를 사용할수 있어야 한다. 그리고 RPM공개열쇠를 포함하는 공개열쇠고리(key ring)를 찾을수 있어야 한다.

기본적으로 RPM은 PGPPATH에서 지시하는 PGP기본설정을 사용한다. PGP가 기본적으로 사용하는 열쇠고리를 가지고있지 않을 때에는 /etc/rpmrc파일에 설정해두어야 한다.

pgp_path는 /usr/lib/rpm대신에 쓰일 경로이름이다. 이 경로에는 제작자의 열쇠고리를 포함해야 한다. 자신이 만든 패키쥐에 서명을 하려면 자신의 공개열쇠와 비밀열쇠 한쌍을 만들어 두어야 한다(PGP지도서를 참고할것).

/etc/rpmrc에 기입해주는것외에도 다음의 사항을 추가해야 한다.

· signature : 서명형식. 현재 pgp만 지원된다.

· pgp_name : 자기의 패키쥐에 서명할 사용자이름을 써준다. 패키쥐제작시에 --sign설정값을 추가한다. 제작자의 입력을 받은후에 패키쥐가 만들어지고 동시에 서명된다.

자료기지 재생성 설정값

rpm자료기지를 다시 만드는 명령은 다음과 같다.

```
rpm —rebuilddb
```

패키쥐에서 나온 원천파일과 스페크파일은 삭제된다.

rpm은 강력한 패키쥐관리자로서 매 쏘프트웨어 패키쥐의 제작, 설치, 문의, 검증, 갱신, 제거 등을 할수 있다. 패키쥐(Package)란 설치할 파일들

과 이름, 판본, 설명 등을 포함하는 패키쥐정보를 가진 보관파일이다. 기본적으로 7가지의 동작방식이 있으며 각기 서로 다른 설정값들을 가진다.

기본적인 7가지의 동작방식이란 설치,문의, 검증, 서명확인, 제거, 제작, 자료기지 재생성 등을 말한다.

1) RPM 파일을 읽는 방법

패키쥐이름은 일정한 규칙으로 되여있다. 실례로 다음과 같은 아파치 RPM패키쥐가 있다고 가정한다.

Apache-1.3.26-i686.rpm

- linux : RPM패키쥐이름이다.
- 2.0.2 : linux 패키쥐의 판본이다.
- i686 : 실행2진의 체계구성방식이다.

2) RPM패키쥐의 설치

보통 RPM을 설치하려면 rpm -ivh설정값을 함께 사용하여 설치하려고 하는 패키쥐명을 지정하게 된다.

```
# rpm -ivh apache-1.3.22-i686.rpm
```

우의 명령으로 쉽게 rpm이 설치되는 경우도 있지만 설치할 때에 이미 있던 파일이나 의존성관계 등으로 인하여 여러가지 오유를 발생하는 경우도 있다.

크게 다음과 같은 3가지의 경우에 해당한 오유발생의 해결방법에 대하여 아래에 설명하였다.

ㄱ) 이미 꼭같은 패키쥐가 설치되여있는 경우

```
kcc@linux:~
#rpm -ivh apache-1.3.22-i686.rpm
apache package apache-1.3.22 Is already installed
error : apache-1.3.22-i686.rpm cannot be installed
```

해결책: --replacepkgs설정값을 설치시에 추가적으로 입력하여 이미 있던 패키쥐를 덧쓰기한다.

ㄴ) 설치될 패키쥐의 파일이 충돌하는 경우

```
kcc@linux:~
#rpm -ivh apache-1.3.22-i686.rpm
apache /usr/xxx.a conflicts with from mod_ssl-2.8.4
error : apache-1.3.22-i686.rpm cannot be installed
```

패키쥐에 포함된 파일의 서고가 있는데 현재 설치하는 패키쥐가 꼭같은 서고의 다른 판본을 설치하려고 하는 경우로서 하나하나 원인을 해결해나가야 한다. -replacepkgs설정값으로 강제설치하는 방법이 있다.

ㄷ) 의존성문제가 발생하는 경우

이 경우에는 해당한 패키쥐를 설치하기전에 먼저 설치해야 하는 패키쥐가 있으므로 오유출력에 나타나는 패키쥐를 먼저 설치하여야 한다.

3) 패키쥐제거

rpm패키쥐제거는 rpm -e설정값과 함께 제거하려는 패키쥐명을 지정한다.
패키쥐제거는 페키지이름만 입력하여도 할수 있다.

```
# rpm -e apache
```

의존성문제에 관한 오유통보문이 나올수도 있다.

```
kcc@linux:~
# rpm -e apache
removing these packages would break dependencies:
linux is needed by bmod_ssl-2.8.4
```

이때에는 의존성이 제기되는 패키쥐를 먼저 삭제하여야 한다. 만일 의존성을 무시하려면 --nodeps설정값으로 삭제하려고 하는 패키쥐만 삭제할수 있

는데 이때에 주의해야 할 필요가 있다.

4) 패키쥐갱신

이미 설치된 패키쥐를 새로운 판본의 패키쥐로 갱신할 때에는 rpm -Uvh 설정값을 사용한다.

```
kcc@linux:~
# rpm -Uvh apache-1.3.26-i686.rpm
linux ########################
```

만일 이전 판본의 패키쥐설치를 하려고 하는 경우에는 -oldpackage설정값을 사용한다.

5) 패키쥐정보확인

```
kcc@linux:~
# rpm -q apache
apache-1.3.22-i686.rpm
```

【관련명령문】

패키쥐가 설치된 경로찾기

rpm -ql설정값과 함께 패키쥐이름을 지정하면 지정한 패키쥐가 설치된 등록부와 파일들을 볼수 있다.

```
kcc@linux:~
# rpm -ql apache
/etc/httpd
/etc/httpd/conf
/etc/httpd/conf/access.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/conf/magic
/etc/httpd/conf/srm.cont
/etc/httpd/logs
/etc/httpd/modules
/etc/logrotate.d/apache
/etc/rc.d/httpd
```

참 고

설치된 파일이나 등록부로 해당 패키쥐를 찾기

설치된 파일이 어떤 패키쥐에 포함되는가 하는 의문이 제기되는 경우가 많다. 보통 이러한 경우에 rpm -qf 설정값과 함께 찾으려고 하는 파일을 절대경로로 지정하면 지정한 파일에 해당하는 패키쥐명을 보여주게 된다.

```
kcc@linux:~
# rpm -qf/etc/httpd/conf/httpd.conf
apache-1.3.22-i686.rpm
```

rpm2cpio /usr/bin/rpm2cpio

RAM변환명령

RedHat패키쥐(RPM)를 cpio파일로 변환하는 명령문이다.

형식 : rpm2cpio[파일이름]

rpm2cpio는 .rpm파일이름을 하나 받아들여 표준출력으로 cpio보관파일을 출력한다. 만약 아무러한 전달인수도 없는 경우에는 표준입력으로부터 rpm 파일을 받아들인다.

```
# rpm2cpio rpm-1.1-1.i386.rpm
# rpm2cpio < glint-1.0-1.i386.rpm
```

rsh /usr/bin/rsh

원격호스트쉘을 사용

원격호스트에서 쉘을 사용하여 명령을 실행할수 있게 하는 명령문이다.

형식 : rsh[설정값]호스트명[명령문]

-l usermame : 호스트에 접근하는 사용자이름을 지정한다.

rsh명령은 원격쉘(remote shell)로서 원격호스트에 명령문을 실행시킨다.

대부분의 r명령은 보안에 미약한데 rsh도 마찬가지이다. rsh명령대신에 ssh명령을 사용해보자. rsh는 지정된 원격호스트에 련결한 다음 사용자가 제공한 명령을 수행하도록 한다. 아래와 같이 원격호스트 linux2에 ls를 한다면 다음과 같이 실행한다.

```
# rsh linux2 "ls -al/"
```

RedHat 7.x판본에서는 보안상의 약점을 방지하기 위한데로부터 /etc/inetd.conf로부터 /etc/xinetd.d/rlogin파일로 설정파일이 변경되였다. 이 파일에서 보면 기본값으로 rsh를 허용하지 않는다.

```
kcc@linux:~

# cat /etc/xinetd.d/rsh
# default: on
# description: The rshd server is the server for the rcmd(3) routine and,\
#   consequently, for the rsh(1) program, The server provides \
#   remote execution facillties with authentication based on \
#   privileged port numbers from frusted hosts.
Service shell
{
      disable=yes
      socket_type          =stream
wait                  = no
    user                  = root
```

```
        log_on_success      += USERID
        log_on_failure      += USERID
        server               = /usr/sbin/in.rshd
}
```

참 고

r명령의 위험성

rcp, rlogin, rsh 등과 같이 원격호스트에서 사용하기 위한 r명령들은 BSD(Berkeley Software Distribution)계렬에서 사용하여 온것들이다.

이것은 망을 통해 원격호스트의 쉘이나 명령을 실행하는것으로서 사용자의 접근권한을 인증하는 과정을 포함한다. 그런데 이 명령들은 원격지에서 작업할수 있게 하는 편리성을 제공하는 대신에 호스트와 식별이름만으로 통과하는 단순한 인증절차로 하여 보안상의 문제를 쉽게 발생시키는 경우가 적지 않다. 보다 위험한것은 《+》문자를 사용하여 임의의 사용자가 접근하는것도 허용할수 있게 하는것이다.

관리자의 rhosts파일에 《++》와 같은 내용을 써넣은 뒤에 rlogin으로 뿌리권한을 얻는 형태의 공격이 지난 시기에 빈번하였다. 때문에 대부분의 체계들에서는 r명령을 안전한 s명령(ssh, scp 등)으로 교체하여 사용하고있다. 물론 현재는 우와 같은 방법의 공격은 거의나 허용되지 않고있다.

rstat

/usr/bin/rstat

체계식별이름정보의 제공

허용된 호스트에 체계식별이름을 보여주는 명령문이다.

rstat는 CPU와 가상주기억기의 사용통계, 망가동시간, 하드디스크 등에 대한 정보를 제공하는 기능을 수행한다. r명령문의 대부분이 보안상의 약점을 가지고있는바 rstat도 례외로 되지 않는다. 그러므로 될수록이면 이 명령을 사용하지 않는것이 좋다.

runlevel

/sbin/runlevel

실행준위의 보기

현재와 이전의 체계실행준위를 찾는 명령문이다.

형식 : runlevel

runlevel명령은 이전체계의 실행준위와 현재체계의 실행준위를 표준출력에 의해 공백문자로 나누어 보여준다. 만일 이전체계의 실행준위가 없으면 N을 표시한다. 이 명령은 System-V의 who -r명령을 대신하여 rc스크립트안에서 사용될수 있다. 또한 init의 새로운 판본에서 이 정보들은 RUNLEVEL과 PREVLEVEL환경변수를 지정하는데 사용될수 있다.

```
kcc@linux:~
# runlevel
N 5
```

이 체계에는 이전체계의 실행준위가 없으며 현재의 실행준위는 5로써 X Window를 기본으로 사용한다. /etc/inittab파일에서 주해(Note)에 의해 매 실행준위에 대한 설명을 보여주는 실례를 아래에 주었다. 해당한 Linux 배포판에 따라 약간 다를수 있지만 /etc/inittab파일을 보면 매 준위가 어떤 것인가를 확인할수 있다.

```
kcc@linux:~
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS(The same as 3, if you do not have networking)
# 3 - full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
```

이것은 init명령과 함께 련관시켜볼 필요가 있다.

rwall

/usr/bin/rwall

통보문보내기

원격허용된 호스트를 사용하는 사용자에게 통보문을 보내는 명령문이다.

형식 : rwall호스트[파일]

rwall은 원격사용자가 망에 있는 모든 사용자에게 통보문을 보내는 명령문이다. 이 명령도 역시 보안에 매우 미약하므로 될수록이면 사용하지 않는 것이 좋다. RedHat 7.X이전의 판본에서는 /etc/inetd.conf에서 이 명령에 의한 봉사를 금지하도록 하며 7.0.X판본에서는 rwalld데몬을 실행하지 않도록 할 필요가 있다. 데몬을 제거한 후 ntsysv편의프로그람으로 기동할 때 자동실행되는 rwalld데몬을 제거하도록 한다.

```
kcc@linux:~
# /etc/rc.d/init.d/rwalld stop
rwho 봉사를 완료중입니다:          [실패]
```

rwho /usr/bin/rwho

접속자확인

허용된 호스트에 현재 접속된 사용자의 정보를 보여주는 명령문이다.

형식 : rwho[설정값]

who명령의 망확장판명령으로서 현재 체계접속(log in)된 사용자의 정보를 보여주는 명령문이다. 허용된 호스트에게 who명령의 결과와 같은 정보를 보여주게 된다.

```
kcc@linux:~
# who
root    :0      Jun 7 09:06
root    pts/1   Jun 9 15:59 (211.49.155.208)
```

이 명령문 역시 보안에 매우 미약하므로 사용하기를 권고하지 않는다. RedHat 7.X이전의 판본에서는 /etc/inetd.conf에서 이 명령에 의한 봉사를 금지하도록 한다. 그리고 7.0.X판본에서는 rwhod데몬을 실행하지 않도록 한다. 데몬을 제거한 다음 ntsysv편의프로그람으로 기동할 때 자동실행되는 데몬에서 제거하도록 한다.

```
kcc@linux:~
# /etc/rc.d/init.d/rwhod stop
rwho 봉사를 완료중입니다.     [실패]
```

S

script /usr/bin/script

말단의 작업기록

말단에서 발생하는 모든 대화(session)를 기록하는 명령문이다.

형식 : script[-a][파일]

-a : 새로 기록하는것이 아니라 이전의 파일에 덧붙인다.

script명령은 말단에서 진행되는 모든 작업을 파일에 보관한다. 따라서 차후에 이전의 작업내용을 확인하려는 경우에 매우 유용하다.

script명령만을 실행하는 경우에는 기본적으로 typescript파일에 말단에서 일어난 작업내용들이 기록보관된다. 만일 script에 파일이름을 지정하면 보관한 파일에 작업내용이 기록된다. -a설정값은 파일을 지정하지 않은 경우에 기본적인 보관파일인 typescript파일에 추가적으로 말단의 작업내용을 기록한다.

실행중인 script명령을 완료하기 위해서는 exit나 Ctrl+D, logout명령을 입력한다. 말단에서 일어나는 작업들을 typescript에 기록보관하는 실례를 아래에 주었다.

```
kcc@linux:~

# script
Script started, file is typescript
# rm -rf tes/
# mkdir mktest
# ls
Desktop     column_txt  mailbak  Office2.0-eval
# exit
exit
Script done, file is typescript
```

script명령으로 작업된 내용을 보려고 할 때에는 작업내용이 기록보관된 typescript파일을 본다.

```
# cat typescript
```

scp /usr/bin/scp

파일복사

원격보안복사명령문(secure copy)이다.

형식 : scp[설정값][user@host:]file1[…]

-p : 원본파일의 변경시간, 접근시간, 권한을 변경함이 없이 그대로 보관한다.
-r : 지정한 등록부의 하위등록부 및 파일까지 모두 복사한다.
-v : 상세한 진행사항을 보여준다.
-F ssh_config : 지정한 설정파일을 사용한다.
-P port : 지정한 포구로 접속시도를 한다.

scp명령은 원격호스트의 파일을 복사하는 보안복사명령문으로서 신분인증 및 암호화를 사용하여 telnet, rlogin, rcp, rsh 등의 명령들을 안전하게 대치할수 있게 한다.

아래의 실례에서는 webmaster식별이름으로 /home/wed master/등록부 아래의 모든 파일을 linuxroot.kp호스트의 현재의 파일등록부에로 복사한다.

```
# scp webmaster@linuxroot.co.kp/home/webmaster/* ./
```

지정한 파일등록부의 아래준위 등록부까지 모두 복사하려면 -r설정값을 사용한다.

아래의 실례에서는 admin식별이름권한으로 linuxroot호스트에 있는 /home/admin/경로의 모든 아래준위 파일등록부와 파일을 국부호스트의 작업파일등록부아래에 있는 backup파일등록부에 모두 복사한다.

```
# scp admin@linuxroot.co.kp:/home/admin/* ./backup
```

setfdprm /usr/bin/setfdprm

플로피디스크설정

플로피디스크의 명세(Specification)를 설정하는 명령문이다.

형식 : setfdprm[설정값]장치명[이름]

-c 장치이름 : /etc/tdprm파일의 환경설정을 지운다.
-n 장치이름 : 초기화검색통보문을 비활성화한다.
-p 장치이름[이름] : 지정한 이름으로 장치의 설정을 영구적으로 바꾸어 버린다.
-y 장치이름 : 초기화검색통보문을 활성화한다.

setfdprm명령은 플로피디스크를 자동검색할수 있도록 디스크명세를 지정하거나 이전의 설정을 지우고 검사통보문을 활성화 혹은 비활성화한다.
fdformat /dev/fd0H1440명령으로 플로피디스크를 초기화한다.

```
kcc@linux:~
# fdformat /dev/fdOH1440
Double-sided, 80 tracks, 18sec/track Total capacity 1440kB.
FormattIng... done
Verifying... done
```

setfdprm명령으로 플로피디스크이름이 1440/1440인 장치를 /dev/fd0 장치이름으로 지정하여 보자.

```
# setfdprm/dev/fd0 1440/1440
```

우의 실례와는 다르게 새로 지정한 장치이름으로 초기화가 가능하다.

```
# fdformat/dev/fd0
```

setup /usr/sbin/setup

체계설정

체계관련명령을 설정하는 안내문방식편의프로그람이다.

형식 : setup

setup명령은 체계관련설정을 안내문방식으로 설정하는 편의프로그람이다.

사용자정보설정, 방화벽설정(ipchains), 건반설정, 마우스설정, 망설정, 인쇄기설정, 기동시에 데몬의 선택, 음향기판설정, 시간대(timezone)조절, 등의 설정을 할수 있다.

안내문들은 다음과 같다.

안내문이름	설 명	직접실행명령문
Authentication configuration	사용자정보설정	authconfig
Firewal configuration	방화벽(ipchains)설정	lokkit
Keyboard configuration	건반설정	kddconfig
Mouse configuration	마우스설정	mouseconhg
Network configuration	망설정	netconfig
Printer configuration	인쇄기설정	printconf-tui
System service	기동시의 데몬설정	ntstysv
Sound card corfiguration	음향기판설정	sunconfig
Timezone configuration	시간대조절	timeconfig
X configuration	X Window설정	Xconfigurator

setup명령으로 이 안내문의 전체를 볼수도 있으며 매 실행명령문을 입력하여 매 안내문상에서의 직접적인 설정도 할수 있다.

sh

/bin/sh

기본쉘의 호출

기본쉘을 호출하는 명령문이다.

sh명령문은 본쉘을 호출한다. /bin/sh는 본쉘을 의미하였지만 현재는 대부분 /bin/bash를 련결(link)하여 사용한다.

다음과 같이 확인해볼수 있다.

```
kcc@linux:~

$ whereis sh
sh: /bin/sh

$ ls -al /bi/sh
lrwxrwxrwx  1 root  root  4 5월 19 16:15 /bin/sh -> bash
```

만일 쉘스크립트로 작성한 ftpdown_count.sh파일이 있는 경우, 이것은 ./명령문이나 sh명령으로 실행이 가능하다.

```
$ sh ftpdown_count.sh
```

자세한 설명에 대해서는 쉘스크립트부분을 참고하면 된다.

shar

/usr/bin/shar

쉘압축편의프로그람

NewsGroup등에서 사용하며 쉽게 내용을 열람할수 있는 압축편의프로그람이다.

shar로 압축된 파일은 #/biN/sh머리부(header)와 문서고(archive)를 풀기 위한 명령으로 구성된 쉘스크립트로 압축되지 않은 파일들이 묶여있는 형태이다. 이 파일은 뉴스집단관리자가 원천파일의 읽기, 정리, 삭제 등을 쉽게 할수 있는것으로 하여 Internet NewsGroup에서 흔히 볼수 있었지만 최근에는 대부분이 tar/gzip로 교체되고있다.

shar문서고는 unshar명령으로 압축을 해제한다.

shar명령으로 압축하기

c원천파일을 shar로 압축해보자.

```
$ shar -o[보관할 파일][원본파일]
```

또는 다음과 같은 방법을 사용할수도 있다.

```
$ shar[원본파일]>[저장할 파일]
```

우와 같은 입력형식으로 shar문서고형태의 압축을 할수 있다.

그러면 hello.c원천파일을 hello.shar파일로 압축해보자.

```
$ sh hello.c > hello.shar
```

shar형태로 된 파일을 보면 쉘스크립트형식으로 되여있다.

```
kcc@linux:~

$ cat hello.shar
# !/bin/sh
# This is a shell archive (produced by GNU sharutils 4.2.1).
# To extract the files from this archive, save it to some FILE, remove
# everything before the '!/bin/sh' line above, then type 'sh FILE'.
#
# Made on  2004-07-14 13:26  KST  by <pirana@localhost>
# Source  directory  was '/home/pirania'.
#
# Existing  files  will *not* be overwritten unless '-c' is  specified.
------------------------------생략------------------------------
```

shar명령으로 압축풀기

hello.shar파일을 해제할 때에는 unshar명령을 사용한다.

```
$ unshar  hello.shar
```

【관련명령문】

unshar : shar파일의 압축해제를 하는 명령문

showmount /usr/bin/showmount

마운트정보보기

체계내부 및 원격으로의 mount정보를 보여주는 명령문이다.

형식 : showmount[설정값][호스트]

-a, —all : 《호스트명:등록부》형식으로 모든 원격mount정보를 보여준다. 《호스트명》은 의뢰기이름이며 《등록부》는 mount된 파일체계등록부의 정보이다.
-d, —directories : 의뢰기에서 원격으로 mount한 등록부의 목록을 보여준다.
-e, —exports : export된 등록부의 목록을 보여준다.
-h, —help : 도움말을 보여준다.
-v, —version : 판본정보를 보여준다.

showmount명령은 NFS/NIS명령으로 지정한 체계에서 원격으로 파일체계를 마운트(mount)한 의뢰기목록들을 보여준다. 만일 호스트를 지정하지 않으면 현재 봉사기에서 마운트(mount)된 파일등록부의 정보를 보여준다.

참 고

점 대 점(point-to-point)

점 대 점이란 두 장치간에 중계장치가 없이 한 장치와 다른 한 장치사이의 직접적인 련결을 말한다. 이 방식은 콤퓨터와 말단간에 계속적으로 대화를 나누며 빠른 응답을 필요로 하는 경우와 콤퓨터체계가 다른 대형콤퓨터에 련결되여 말단처럼 사용되는 경우에 주로 리용된다.

shutdown /usr/bin/shutdown

체계끝내기

정해진 지시사항에 따라 체계의 프로세스를 끝내고 주기억기의 내용을 보관한 다음 안전하게 끝내도록 하는 명령문이다.

형식 : shutdown[설정값]시간[경고통보문]

-c : 진행중의 shutdown지령을 취소한다.
-f : 재기동시에 fsck지령을 건너뛰면서 빠른 재기동을 한다.
-h : shutdown이 완료된 후에 체계를 끝낸다.
-k : 경고통보문만을 보여주고 실제적인 끝내기(shutdown)를 진행하지는 않는다.
-n : shutdown을 위해 init를 호출하지 않고 끝내기를 진행한다.
-r : 체계를 끝낸 다음에 재기동한다.
-t sec : 지정한 시간에 체계를 재기동한다.

시간
now : 현재 즉시 끝낸다.
+m : 지정한 m분이후에 끝낸다.
hh : mm : 지정한 몇시(hh) 몇분(mm)에 완료한다.

shutdown명령은 체계를 안전하게 끝내게 하는 명령문이다. 현재 접속중인 모든 사용자는 체계가 완료된다는 통보문을 받고 망접속작업을 끝내게 된다. 체계를 즉시에 끝내기 위해 다음과 같이 실행할수 있다.

```
# shutdown -h now
```

우의 명령은 halt, init 0명령과 류사한 기능이다.
체계완료후에 재기동하기 위해서는 다음과 같이 실행할수 있다.

```
# shutdown -r now
```

이 명령은 reboot, init 6명령과 같다.

일정시간이 지난 다음 체계가 완료되는것을 취소하려고 할 때에는 shutdown -c명령을 사용한다.

slattach

/usr/bin/slattach

직렬회선련결

직렬회선을 망대면부로 련결하는 명령문이다.

형식 : slattach [설정값][tty]

-c commad : 접속이 끊어질 때 지정한 command를 실행한다.
-d : 오유수정방식
-h : 접속이 끊어질 때 빠져나온다,
-q : 어떤 통보문도 보여주지 않는다. (quiet mode)
-l : /var/lock에 UUCP 류형의 lock파일을 생성한다.
-n : mesg n 지령과 동일하다.
-m : 8bit방식으로 초기화하지 않는다.
-e : 초기화한 후 즉시 빠져나온다.
-L : 3선작용을 활성화한다.
-p proto : 지정한 proto로 통신규약종류를 선택한다. 여기에는 slip, adaptive, ppp, kiss 등이 있다.
-s speed : 지정한 speed 로 속도를 지정한다.

slattach명령은 TTY회선을 망대면부에 지정하는 역할을 하여 점 대 점(point to point)접속으로 다른 콤퓨터를 리용할수 있게 한다. SLIP망대면부를 직접 tty 1포구에 련결할수 있다.

```
# sattach /dev/tty 1
```

sleep /usr/bin/sleep

대기

지정한 시간동안 아무런 동작도 하지 않고 대기하는 명령문이다.

형식 : sleep [설정값]수자[단위]

—help :표준출력으로 사용법을 출력한다.
—version : 표준출력으로 판본정보를 출력한다.

sleep명령은 쉘에서 사용하는 wait순환고리와 같은 기능을 수행한다. 지정한 시간(기본은 초단위시간)동안 아무런 작업도 하지 않고 대기상태로 된다. 이 명령은 시간지정(timing), 배경(background) 등에서 특정한 시간이 되기를 기다릴 때에 유용하게 사용될수 있다.

지정가능한 기본적인 수값단위는 초이다. 아래와 같은 수값단위로 원하는 시간을 지정하여준다.

수값단위	설 명
S	초
M	분
H	시간
D	일

slocate

/usr/bin/slocate

파일찾기

자체자료기지를 통해 파일을 찾아주는 locate명령문의 보안강화판명령이다.

형식 : slocate[설정값]

-u : /경로에서부터 slocate자료기지를 만든다

-U dir : 지정한 등록부경로에서부터 slooate자료기지를 만든다.

-c : -u 또는 -U설정값을 사용하는 경우, /etc/updatedb.conf의 위치를 추적한다.

-e dir1,dir2,.. : -u 또는 -U설정값을 사용하는 경우, 지정한 등록부들을 제외한다.

-f fs_type1,.. : -u 또는 -U설정값을 사용하는 경우, 지정한 파일체계를 제외한다.

-l level : 보안준위가 0이면 보안검사를 하지 않는 대신에 검색이 보다 빠르다. 한편 1이 기정값으로서 이때에는 보안검사를 한다.

-q : 오유통보문을 보여주지 않는다.

-n num : 지정한 수자만큼의 결과만을 보여준다.

-i : 검색에서 일일이 확인한다.

-r regexp : POSIX표현방식으로 자료기지를 검색한다.

-o file : 지정한 파일명으로 자료기지파일을 만든다.

-d path : 지정한 자료기지경로에서 찾는다.

-h, —help : 사용법을 보여준다.

-v, —verbose : 자료기지생성때에 상세통보문을 보여준다.

-V, —version : 판본정보를 보여준다.

체계가 자동적으로 slocate자료기지를 갱신하지 않으면 파일이나 파일등록부정보를 제대로 확인할수 없다. 때문에 체계에서는 updatedb프로그람을 하루에 한번씩 cron으로 실행하여 자료기지를 갱신한다.

RedHat배포판인 경우 하루에 한번씩 실행하는 cron설정등록부인 /etc/cron.daily/slocate.cron파일을 보면 /usr/hin/updatedb파일이 실행되는 것을 확인할수 있다.

```
kcc@linux:~
# cat /etc/cron.daily/socate.cron
#!/bin/sh
 renice +19 -p  $$ >/dev/null  2>&1
/usr/bin/updatedb -f "nfs,smbfs,ncpfs,proc,devpts" -e
"/tmp,/var/tmp,/usr/tmp,/afs,/net"
```

find명령과는 달리 slocate명령은 자료기지에 미리 생성하며 파일찾기를 빠른 속도로 할수 있다. httpd.conf파일을 slocate명령으로 찾는 례를 아래에 보여주었다.

```
kcc@linux:~
# slocate  httpd.conf
warning: slocate: warning: database /var/lib/slocate/sloca te.db'
 is more than 8 days old
/etc/httpd/conf/httpd.conf
/usr/share/doc/mod_perl-1.24_01/eg/httpd.conf.pl
/usr/share/apacheconf/httpd.conf.xsl
/hdb2/oracle/product/9.0.1/Apache/Apache/conf/httpd.conf.tmp
/hdb2/oracle/prroduct/9.0.1/Apache/Apache/conf/httpd.conf.default
/hdb2/oracle/product/9.0.1/Apache/Apache/conf/httpd.conf
/hdb2/oracle9i/Disk1/stage/Components/oracle.apache.apache/1.3.12.0.0c/
1/DataFiles/Expanded/instantiate_unix/httpd.conf.tmp
```

우의 경고통보문에서는 /var/lib/slocate/slocate.db자료기지파일이 8일동안 갱신되지 않아 정확한 정보로 되지 않을수 있다는것을 보여준다.

그러므로 crond데몬으로 하루에 한번씩 갱신할것을 권고한다.

【관련명령문】

find　:　주어진 조건에 따라 등록부를 검색하여 해당한 파일을 찾는 명령문

locate :　자체자료기지를 통해 파일을 찾아주는 명령문

sort

/usr/bin/sort

내용정렬

본문파일의 내용을 영문자모순서로 정렬하는 명령문이다.

형식 : sort[설정값][파일]

정렬설정값

-b, —ignore-leading-blanks : 공백을 무시한다.

-d, —dictionary-order : 공백과 자모문자의 순서를 비교한다.

-f, —ignore-case : 모든 문자를 소문자로 인식한다.

-g, —general-numeric-sort : 수자값을 비교정렬한다.

-i, —ignore-nonprinting : 인쇄가능한 문자만 비교한다.

-M, —month-sort : 날자(월)정렬을 compare (unknown) 〈 JAN' 〈 ... 〈 DEC' 순서로 한다.

-n, —numeric-sort : string numerical value값을 비교하여 정렬한다.

-r, —reverse : 비교결과를 역변환한다.

다른 설정값

-c, —check: 입력된 파일이 정렬되였는가 안되였는가를 검사한다.

-k, — key = pos1[,pos2] : key를 pos1에서 시작하여 pos2까지 지정한다.

-m, —merge : 여러개의 입력파일들을 병합한다.

-o, —output=file : 지정한 file에 결과를 보관한다.

-S, —buffer-size=size : 지정한 size크기로 주기억기완충기(main memory buffer)를 사용한다.

-t, —field-separator=sep : 지정한 sep로 field식별자를 지정한다.

-T, — temporary-directory = dir : 기본림시등록부인 $TMPDIR나 /tmp를 사용하지 않고 지정한 dir를 사용한다.

-u, —unique : field내의 동일한 값을 제거한 유일한 값만을 결과값으로 보여준다.

—help : 도움말을 보여준다.

—version : 판본정보를 보여준다.

sort명령은 지정한 파일로부터 입력받아 행들을 분류하고 그 결과를 표준출력으로 보여준다. 만일 입력파일이 2개이상이면 파일들을 병합하여 하나의 파일로 분류한다.

다음의 실례에서는 /etc등록부의 내용을 정렬하여 보여준다.

```
kcc@linux:~

# ls /etc/  |  sort  |  more
CORBA
DIR_ COLORS
FreeWnn
Muttrc
X11
a2ps-site.cfg
a2ps.cfg
adjtlme
alchemist
aliases
aliases.db
aliases.rpmnew
aliases.rpmsave
amanda
amandates
amd.conf
amd.net
anacrontab
at.deny
auto.master
auto.misc
bashrc
--More--
```

만일 어떤 field안의 중복되는 행을 화면출력하지 않도록 하려는 경우에는 아래의 실례에서와 같이 -u설정값을 사용한다. 이것은 uniq명령과 같은 기능을 수행한다.

```
kcc@linux:~

# cat sortfile
1234
1234
1233

1222
1234
1222
12345

# sort -u sortfile

1222
1233
1234
12345
```

만일 파일안의 대소문자의 구분을 없애려는 경우에는 -f설정값을 사용한다.

```
kcc@linux:~

# cat sortfile
111
122
123
admIn@ l inuxroot.co.kp
ADmin@ l inuxroot.co.kp
ADMIN@LINUXROOT.CO.KP
```

아래와 같이 -f설정값을 사용하여 전자우편주소를 같은 내용으로 보여준다.

```
kcc@linux:~

# sort -fu sort
111
122
123
admin@linuxroot.co.kp
```

source

내부명령문

환경설정의 반영

스크립트나 환경파일을 읽어들여 환경변수를 변경한 경우에도 직접 반영할 수 있게 하는 명령문이다.

형식 : source파일

source명령은 환경설정파일이나 스크립트내의 환경변수값을 변경한 다음 쉘상에서 변경된 환경변수를 읽어들일수 있다.

만일 아래와 같이 Oracle을 설치하기 위한 환경변수를 $HOME/.bash_profile에 추가하였다고 하자.

```
kcc@linux:~

$ cat $HOME/.bash_ profile
export ORACLE_ BASE=/hdb1/oracle
export ORACLE_ HOME=/hdb1/oracle/816
export ORACLE_ SID=RAC1

export NLS_ LANG=AMERICAN_ AMERICA.KO16KSC5601
export TNS ADMIN=$ORACLE_HOME/network/admin
export ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOM
E/lib:$ORACLE_HOME/jdbc/lib
```

다음과 같이 입력하면 변경된 환경변수를 즉시 읽어들일수 있다.

```
$ source $HOME/.bash_ profile
```

이 source명령은 파일이름으로부터 명령을 읽어 현재의 쉘환경내에서 실행한다.

split

/usr/bin/split

파일분할

디스크를 사용하거나 우편을 보낼 때 적당한 크기로 파일을 나누는 명령문이다.

형식 : split[설정값][INPUT[PREFIX]]

-b, —bytes=SIZE : 지정한 SIZE바이트만큼 파일을 만든다.
—help : 도움말을 보여준다.
—version : 판본정보를 보여준다.

split명령은 파일을 작은 쪼각으로 나누어주는 편의프로그람으로서 플로피디스크에 예비보관을 하려고 하거나 전자우편의 첨부파일로 사용할 때 파일을 나누어 올리적재(upload)를 하려고 할 때에 유용하다.

아래의 실례는 현재등록부의 모든 파일을 압축하는데 5MB크기로 split.tar.gz이름으로 파일을 만드는것이다. 5MB가 넘으면 다른 이름의 파일로 분할압축된다.

```
kcc@linux:~
# tar cvzf - * | split -b 5m - split.tar.gz
2002sheet0417_1.jpg
2002sheet0417_10.jpg
2002shee10417_11.jpg
 ---중간 생략------------
2002sheet0417_14.jpg
2002sheet0417_15.jpg
2002sheet0417_7.jpg
2002sheet0417_8.jpg
2002sheet0417_9.jpg
sheet_liunuxflash04.hwp
```

압축파일의 용량이 5MB를 넘으면 split.tar.gz파일의 이름뒤에 aa가 붙으며 다시 split.tar.gzab에 뒤이어 압축하게 된다. 이와 같은 방식으로 ac, ad도 볼수 있다.

```
kcc@linux:~

# ls
2004sheet0417_1.jpg    2004sheet0417_26.jpg  2004sheet0417_41-1.jpg
2004sheet0417_10jpg    2004sheet0417_27.jpg  2004sheet0417_41-2.jpg
2004sheet0417_11.jpg   2004sheet0417_28.jpg  2004sheet0417_41.jpg
―――――――――――――――――――― 중간 생략 ――――――――――――――――――
2004sheet0417_2.jpg   2004sheet0417_35.jpg  2004sheet0417_9.jpg
2004sheet0417_20.jpg  2004sheet0417_36.jpg  sheet_linuxflash04.hwp
2004sheet0417_21.jpg  2004sheet0417_37.jpg  split.tar.gzaa
2004sheet0417_22.Jpg  2004sheet0417_38.jpg  split.tar.gzab
2004sheet0417_23.jpg  2004sheet0417_39.Jpg  split.tar.gzac
2004sheet0417_24.jpg  2004sheet0417_4.jpg   split.tar.gzad
2004sheet0417_25.jpg  2004sheet0417_40.jpg
```

분리된 파일을 하나의 파일로 합치기 위해서는 cat명령을 사용한다.

```
# cat split.tar.gza* >> split.tar.gz
```

참 고

KDE(K deskttop environment)

KDE는 UNIX전용작업기용의 공개원천도형탁상(open source graphic desktop)환경이다. KDE는 초기에 《Kool desktop environmet》라고 불리웠으며 인터네트상의 공식적인 KDE mailing list와 많은 수의 News Group과 1RC chanel 등에서의 활발한 협조를 통해 현재에도 개발진행중에 있는 프로젝트이다. KDE는 완전한 GU1와 함께 파일관리자(File Manager), 윈도우관리자(Window Manager), 도움말체계, 구성관리체계, 도구 및 편의프로그람, 그리고 여러개의 응용프로그람 등을 포함하고있다. KDE응용프로그람들가운데서 가장 유명한것은

Koffice인데 여기에는 문서편집프로그람, 표처리프로그람, 연시물편집프로그람, 벡토르도형처리프로그람, 화상편집프로그람 등이 포함되여있다.

KDE는 현재 Linux, Solaris, FreeBSD OpenBSD 및 LinuxPPC 등에서 사용되고있으며 세계적으로 많은 프로그람전문가들이 KDE개발에 협조하고있다.

ssh /usr/bin/ssh

SSH의뢰기

암호화에 의해 안전한 접속을 제공하는 원격 SSH봉사기에 접속하기 위한 의뢰기프로그람이다.

SSH는 Secure SHell의 략자로서 암호화되고 압축된 련결을 제공한다. 이전의 Telnet는 봉사기와 련결된 상태에서의 파케트송수신이 모두 망에 로출되여 내용이 도청당하기 쉬운 부족점이 있었다. 이러한 문제점은 ssh를 사용함으로써 해결할수 있다.

```
$ ssh -l admin liuxroot.co.kP
```

또는 다음과 같이 입력할수 있다.

```
$ ssh admin@linuxroot.co.kp
The authenticity of host linuxroot.co.kp(61. 40.233.122)
can't be established
RSA key fingerprint is 39:ff: a3:70:bd:20:16:a6:7c:
5e:d4:7a:60:37:0c:f7.
Are you sure you want to continue connecting(yes/no)?
```

처음 SSH봉사기에 접속하도록 하기 위해 우와 같은 통보문을 내보낸다. 《yes》를 입력하면 RSA열쇠를 자기의 국부환경체계의 $HOME/.ssh/known_hosts파일에 보관한다. 다음에 원격말단과 같이 접속하여 사용하면 된다.

【관련명령문】

scp : ssh봉사기에서 원격으로 복사할수 있는 명령문

stat

/usr/bin/stat

상태보기

파일이나 파일체계의 상태를 보여주는 명령문이다.

형식 : stat[설정값]파일이름

-l : 련결파일의 원천파일정보를 보여준다.
-f : 파일명이 있는 파일체계의 정보를 보여준다.
-v : 판본정보를 보여준다.
-f : 간략화된 정보를 보여준다.

stat명령은 파일이나 파일체계의 크기와 블로크, IO블로크, 접근날자, 수정날자 등을 볼수 있게 한다.

아래와 같이 stat04.hwp파일을 보면 파일에 관한 크기와 블로크, IO블로크, 장치 및 inode 등의 자세한 정보를 보여준다.

```
kcc@linux:~

# stat stat04.hwp
  Flie : "stat04.hwp"
  Size: 4370732   Blocks: 8568   IO Block: 4096   Regular File
Device : 308h/776d  Inode: 980856  Links: 1
Access: (0644/-rw—r—r—)  Uid: ( 0/ root) Gid: ( 0/ root)
Access:  Sun Jun 9 23:58:44 2004
Modify: Wed Apr 17 18:55:56 2004
change:  Wed Apr 17 18:55:56 2004
```

-f설정값은 파일이름이 있는 파일체계의 정보를 보여준다.

```
kcc@linux:~

# stat -f stat04.hwp
  Flie: "stat04.hwp"
  ID: 0   0 Namelen: 255          Type: EXT2/EXT3
Blocks:  Total: 2051703   Free: 859403   Available: 755182   Size: 4096
Inodes:  Total: 1042432   Free: 830200
```

파일등록부를 지정하여 등록부의 상태를 볼수도 있다.

```
kcc@linux:~

# stat /
  File: "/"
  Size: 4096      Blocks: 8     IO Block: 4096   Directory
Device: 301h/769d  Inode: 2     Links: 19
Access: (0755/drwxr-xr-x) Uid: ( 0/ root)  Gid: ( 0/ root)
Access: Sun  Jun 14 10:10:44 2004
Modify: Thu  Jul 11 02:19:32 2004
Change: Thu  Jul 11 02:19:32 2004
```

참 고

Linux에서 만들수 있는 최대크기의 파일용량

이것은 사용하는 파일체계에 따라 다를수 있다. 이전에 사용하던 minix나 ext파일체계에서는 2GB, ext2파일체계에서는 64GB크기까지 만들수 있었다. ext2의 inode는 12개의 블로크와 직접 련결되여 한개는 간접, 한개는 2중, 한개는 3중으로 련결되여있으며 ext2 zone은 256개까지의 zone pointer를 사용할수 있다.

최근의 journaling파일체계에서는 파일크기의 제한이 대폭 완화되였다. 널리 사용되고있는 ext3의 경우에 4TB크기까지의 파일을 만들수 있으며 Reiser나 XFS에서는 그 이상의 크기도 만들수 있다.

strace /usr/bin/strace

프로그램에서 실행되는 프로세스의 추적

체계호출과 신호를 추적하여 프로그램의 실행과정을 볼수 있게 하는 명령문이다.

형식 : strace[설정값]명령문[매개변수]

strace명령은 체계호출과 신호를 추적하여 진단하고 오유수정해주는 도구로서 프로그람에서 실행되는 프로쎄스가 어떤것인가를 추적할수 있게 한다.

이외에도 2진파일에 어떤 경로이름이 번역되여있는가를 보거나 조건을 보려는 경우 혹은 2진파일의 동작원리를 알아보려는 경우에도 사용할수 있다. 아래 실례와 같이 strace명령을 실행해보자. 이것은 ls명령에 대한 다양한 번역경로나 조건, 동작에 대한 정보를 볼수 있게 한다.

```
kcc@linux:~

$ strace ls
execve("/bin/ls", ["ls"], [/* 27 vars*/]) = 0
uname({sys="Liux", node="localhost",...}) = 0
brk(0)                              =0x80579e4
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x40016000
open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such
fileor directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG10644, st_size=70652,…}) = 0
old_mmap(NULL, 70652, PROT_READ, MAP_PRIVATE, 3, 0)
= 0x40017000
close(3)                  = 0
open("/lib/libtermcap.so.2", O_RDONLY) = 3
----------------------------중간생략------------------------
```

strings /usr/bin/strings

문자렬찾기

오브젝트파일 또는 2진파일에서 정상적으로 출력할수 있는 문자렬을 찾는 명령문이다.

형식 : string[설정값]파일

-, -a, —all : 오브젝트파일의 전체를 검색한다.

-f, —print-file-name : 각 문자렬전에 파일이름을 보여준다.

-min-len, -n min-len, —bytes=min-len : min-len수값으로 최소문자렬의 길이를 지정한다. 기본값은 4이다.

-o : -f설정값과 비슷하다.

-t base, —radix=base : 지정한 base형식으로 각 문자렬앞에 편위값(offset)을 보여준다.

다음의 base형식이 지정가능하다.

d : 편위값을 10진수로 기록한다.

o : 편위값을 8진수로 기록한다.

x : 편위값을 16진수로 기록한다.

—target=format : 지정한 형식을 체계의 기본코드형식으로 사용한다.

-v, —version : 판본정보를 보여준다.

strings명령은 오브젝트파일 또는 2진파일에서 인쇄가능한 문자렬을 찾는다. 기본문자렬은 4문자이다.

먼저 hello.c원천파일을 cat명령으로 보자. cat명령은 원천파일의 전체를 그대로 보여준다.

```
kcc@linux:~

$  cat hello.c
# Include <stdio.h>
main( )
{
printf ("tesi");
}
```

hello.c파일을 strings명령으로 다시 한번 보자.

아래의 결과와 같이 기본문자렬이 4문자이상인것만을 검색하여 보여준다.

```
kcc@linux:~
$ strings hello.c
#include <stdio.h>
main()
printf ("test");
```

기본문자렬대신에 지정한 문자렬을 기준으로 문자렬을 찾을수 있다.

실례로 17문자이상만을 찾아보자. 최소문자렬을 지정하기 위해서는 -17, -n 17, --bytes=17 등의 어느 하나로 지정해줄수 있다.

한행의 길이가 18문자인 아래행만을 결과로써 볼수 있다.

```
kcc@linux:~
$ strings -17 hello.c
#include <stdio.h>
```

참 고

Baud

Baud(보드)는 자료전송속도를 나타내는 단위로써 회선을 통하여 1초동안에 전달되는 자료비트의 수를 나타낸다. bps단위와 함께 모뎀의 전송속도에 많이 쓰인다. bps는 초당 비트수로써 만일 3개의 비트가 한 신호단위를 이루는 경우에 보드속도는 bps속도의 1/3로 된다.

strip /usr/bin/strip

목적파일의 크기축소

목적파일(object file)에서 기호(symbol)를 제거한다.

형식 : strip[설정값]오브젝트파일이름

-F bfdname, —target=bfdname : 입력파일을 지정한 bfdname으로 한다.

-O format, —output-target=format : 출력파일을 지정한 형식으로 한다.

-R section, —remove-section=section : section을 삭제한다.

-s, -strip-all : 모든 기호을 제거한다.

-S, -g, —strip-debug : debug symbol만을 제거한다.

—strip-unneeded : 위치재지정정보외의 모든 기호을 지운다.

-K 기호이름, -keep-symbol=기호이름 : 원천파일에서 지정한 기호이름은 삭제하지 않는다.

-o 파일명 : 지정한 목적파일을 strip하여 다른 파일로 보관한다.

-x , —discard-all : global symbol이 아닌것은 지운다.

-x, —discard-locals : 번역기의 위치정보를 지운다.

-v, —verbose : 스트립과정을 자세히 보여준다.

strip는 목적파일의 기호(symbol)를 제거하여 파일의 용량을 줄인다. 다양한 설정값에 의해 선택적으로 기호를 삭제할수 있다. strip는 회선번호정보, 위치재지정정보, debug단락, typchk단락, 주해단락, 파일머리부(file header) 및 XCOFF목적파일의 모든 기호표 또는 일부를 제거할수 있다. 기호가 제거된 목적파일은 실지 실행에서 별다른 이상이 없다. debug설정값 등이 모두 삭제되므로 마지막으로 실행파일이 작성된 상태에서 사용한다.

```
kcc@linux:~

# ls -al
-rwxr-xr-x  1 root   root  5761641  7월 14 15:08 libqt.so.2.3.2
```

file명령문으로 현재 파일이 strip되였는가, 안되였는가를 알수 있다.

파일정보의 끝부분에 보여주는 《not stripped》에 의해 strip가 안된 파일상태를 확인할수 있다.

```
kcc@linux:~

# file libqt.so.2.3.2
libqt.so.2.3.2: ELF 32-bit LSB shared object, Intel 80386, version 1, not stripped
```

libqt.so.2.3.2파일을 strip하여 용량을 줄여보자. -v는 스트립결과를 보여주며 o는 strip된 파일을 다른 파일로 보관한다.

5.7MB정도의 파일용량이 5.0MB정도로 줄어들었다.

```
kcc@linux:~

# strip libqt.so.2.3.2 -vo libqt.so.2.3.2_strip
copy from libqt.so.2.3.2(elf32-i386) to  teststrip(elf32-i386)

# file libqt.so.2.3.2_strip
libqt.so.2.3.2_strip: ELF 32-bit LSB shared object, Intel  80386,
version 1, stripped
《stripped》에 의해 libqt.so.2.3.2_strip파일이 strip되였음을 확인할수
있다.

# ls -al
-rwxr-xr-x  1 root  rcot 5761641  7월 14 15:08  libqt.so.2.3.2
-rw-r — r — 1 root root 5064576 7월 14 15:25 libqt.so.2.3.2_strip
```

참 고

목적코드(object code)

CPU로 직접 실행할수 있는 기계어코드이다. 고급언어로 작성된 원천프로그람은 번역기에 의해 기계어로 변환된다.

stty

/usr/bin/stty

말단환경설정

말단행설정을 확인하고 수정하는 명령문이다.

형식 : stty[설정...]
stty[설정값]

조종설정

[-]parenb : 출력에서 기우성비트를 발생하며 입력에서 기우성비트를 기대한다.

[-]parodd : 홀수검사방식으로 설정(-는 짝수검사방식)

cs5 cs6 cs7 cs8: 문자의 크기를 5, 6, 7, 8bit로 설정한다.

[-]hupcl [-]hup : 마지막프로세스가 tty를 닫으면 Hangup신호를 보낸다.

[-]cstopb : 문자당 2개의 정지비트를 사용한다(-는 한개의 정지비트)

[-]cread : 입력을 받아들인다.

[-]clocal : 모뎀조종신호를 불가능으로 설정한다.

[-]crtscts (np) : RTS/CTS주고받기를 설정한다.

입력설정

[-]ignbrk : break를 무시

[-]brkint : break가 새치기신호를 발생하게 한다.

[-]ignpar : 기우성오유를 무시한다.

[-]parmrk : 기우성오유를 표시한다(255-0-문자순서로).

[-]inpck : 입력기우성검사를 실행한다.

[-]istrip : 입력문자의 상위비트(8번째 비트)를 지운다.

[-]inlc : 개행문자를 복귀(CR:Carriage Return)문자로 번역한다.

[-]igncr : 복귀(CR)문자를 무시한다.

[-]icrnl : 복귀(CR)문자를 개행문자로 번역한다.

[-]ixon : XON/XOFF흐름조종을 설정한다.

[-]ixoff [-]tandem : 체계의 입력완충기가 거의 차면 정지문자를 보내도록 한다. 그후 입력완충기가 비면 시작문자를 보낸다.

[-]iuclc (np) : 대문자를 소문자로 번역한다.

[-]ixany (np) : 어떤 문자이든 출력을 다시 시작할수 있게 허용한다(-는 시작문자만을 허용)

[-]imaxbel (np) : 입력완충기가 찬 상태에서 문자가 도착하면 경고음을 내고 입력완충기에로의 전송을 대기하도록 한다.

출력설정

[-]opost : 후처리출력.

[-]olcuc(np) : 소문자를 대문자로 번역한다.

[-]ocrnl(np) : CR문자를 개행문자로 번역한다.

[-]onlcr (np) : 개행문자를 복귀-개행문자로 번역한다.

[-]onocr (np) : 첫번째 column에서는 복귀문자를 출력하지 않는다.

[-]onlret (np) : 개행문자가 복귀를 수행한다.

[-]ofill (np) : 지연시간timing대신에 채움문자를 사용한다.

[-]ofdel(np): 빈문자(Null)대신에 채움문자로서 Delete문자를 사용한다.

nl1 nl0(np) : 개행문자지연방식.

cr3 cr2 cr1 cr0 (np) : 복귀지연방식.

tab3 tab2 tab1 tab0 (np) : 수평tab지연방식.

bs1 bs0 (np) : 뒤로 지우기(backspace)지연방식.

vt1 vt0 (np) : 수직tab지연방식.

ff1 ff0 (np) : 용지이송(form feed)지연방식.

지역설정

[-]isig : 새치기, 완료, 중지 등의 특수문자를 사용가능.

[-]icanon : erase, kill, werase, rprnt 등의 특수문자를 사용가능.

[-]iexten : 비POSIX특수문자를 사용가능.

[-]echo : 입력문자를 반향(echo)한다.

[-]echoe, [-]crterase : erase문자를 backspace-space-backspace로 반향한다.

[-]echok : kill문자뒤에 개행문자를 표시한다.

[-]echonl : 다른 문자는 반향하지 않아도 개행문자를 반향한다.

[-]noflsh : 새치기나 완료문자 다음 출력을 금지하도록 한다.

[-]xcase (np) : icanon이 설정되여있을 때 입력, 출력에 있어서 대문자를 해당한 문자앞에 \(역사선)을 붙여 사용할수 있게 해준다.

[-]tostop (np) : 말단에 쓰기를 시도하려는 배경(background)작업을 멈추게 한다.

[-]echoprt [-]prterase (np) : \와 /사이에서 지워진 문자를 다시 반향한다.

[-]echoctl [-]ctlecho (np) : 조종문자를 문자그대로가 아니라 ^표기법(실례로 ^c)으로 반향한다.

[-]echoke[-]crtkill (np) : echoctl과 echok설정대신에 echoprt, echoe설정값을 사용하여 한행의 매 문자를 지움으로써 kill특수문자를 반향한다.

조합설정

[-]evenp [-]parity : parenb -parodd cs7과 같다. -을 사용하면 -parenb cs8과 같다.

[-]oddp : parenb parodd cs7과 같다. -을 사용하면 -parenb cs8과 같다.

[-]nl : -icml -onlcr와 같다. -을 사용하면 icrnl -inlcr -igncr onlcr -onlcr -ocrnl -onlret와 같다.

Ek : erase, kill특수문자를 원래의 값으로 되돌린다.

sane : cread -ignbrk brkint -inlcr -igncr icrnl -ixoff -iuclc -ixany imaxbel opost -olcuc -ocrnl onlcr -onocr -onlret -ofill-ofdel nl0 cr0 tab0 bs0 vt0 ff0 isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt echoctl echoke와 같으며 또한 모든 특수문자를 원래의 값으로 되돌린다.

[-]cooked : brkint ignpar istrip icrnl ixon opost isig icanon과 같으며 min, time문자가 같으면 eof, eol문자도 원래의 값으로 되돌린다. -를 사용하면 raw와 같다.

[-]raw : -ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr -icrnl -ixon -ixoff -iuclc -ixany -imaxbel -opost -isig -icanon -xcase min 1 time 0과 같다. -를 사용하면 ccoked와 같다.

[-]cbreak : -icanon과 같다.

[-]pass8 : -parenb -istrip cs8과 같다. -를 사용하면 parenb istrip cs7과 같다.

[-]litout : -parenb -istrip -opost cs8과 같다. -를 사용하면 parenb istrip opost cs7과 같다.

[-]decctlq (np) : -ixany와 같다.

[-]tabs (np) : tab0과 같다. -를 사용하면 tab3과 같다.

[-]lcase [-]LCASE (np) : xcase iuclc olcuc와 같다.

crt : echoe echoctl echoke와 같다.

dec : echoe echoctl echoke -ixany와 같으며 또한 새치기문자를 Ctrl-C로, erase를 Del로, kill을 Ctrl-u로 설정한다.

특수문자

특수문자의 기본값은 체계마다 다르며 《이름값》이라는 문법으로 설정된다. 여기에서 《이름》은 아래에 보여주었으며 《값》은 ^표시법(례하면 ^c) 또는 16진수를 의미하는 0x, 8진수를 나타내는 O, 또는 10진수 등으로 표기할수 있다. 값에다가 ^-을 부여하거나 《undef》하는 경우에 그 특수문자는 사용할수 없게 된다.

intr : 새치기신호를 보낸다.

quit : 완료신호를 보낸다.

erase : 마지막문자를 지운다.

kill : 현재의 한행을 지운다.

eof : 파일끝이라는것을 알린다(입력완료).

eol: 한행의 끝.

eol2(np) : 한행을 마치기 위한 별도의 문자.

swtch (np) : 다른 쉘계층으로 절환

start : 멈추었던 출력을 다시 시작한다.

stop : 출력을 중단한다.

susp : 말단정지신호를 보낸다.

dsusp (np) : 입력을 중지한 후 말단정지신호를 보낸다.

rprnt (np) : 현재행을 다시 인쇄한다.

werase (np) : 마지막단어를 지운다.

lnext (np): 특수문자인 경우에도 다음의 문자는 그대로 입력한다.

특수설정

min N : -icanon이 설정되였을 때 time값이 지날 때까지 한번의 읽기를 만족시킬수 있는 문자의 최소개수를 정한다.

time N : -icanon이 설정된 경우, min개수의 문자가 입력되지 않았을 때의 초과시간을 1/10초단위로 설정한다.

ispeed N : 입력속도를 N으로 설정한다.

ospeed N :출력속도를 N으로 설정한다.

rows N (np) : 핵심부에게 말단이 N행을 가지고있다고 알린다.

cols N columns N (np) : 핵심부에게 말단이 N렬을 가지고있다고 알린다.
size (np) : 핵심부가 인식하고있는 말단의 행렬수를 출력한다.
line N (np) : 회선조종 N을 사용한다.
speed : 말단속도를 표시한다.
N : 입력/출력속도를 N으로 설정한다. N는 다음값들중의 하나로 지정할 수 있다. 0, 50, 75, 110, 134, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, exta, extb 등이다. exta는 19200과 같으며 extb는 38400과 같다. -clocal이 설정되여있는 경우에 0값을 지정하면 회선을 정지시킨다.
-a -all : 사람이 읽을수 있는 형태로 현재설정을 출력한다.
—help : 표준출력으로 사용법을 출력하고 정상완료한다.
-g, —save : 현재의 설정값을 되돌리기 위해 다른 stty에게 인수로 넘기기 위한 형태로 현재설정값을 출력한다.
—version : 표준출력으로 판본정보를 출력하고 정상완료한다.

stty명령은 아무런 인수도 주어지지 않으면 baud, 회선조종번호, stty sane 등으로부터 변화한 회선설정값들을 보여준다.

```
kcc@linux:~
# stty
speed 38400  baud; line = 0;
erase = ^H;
-brkint -imaxbel
```

권한의 읽기/설정은 표준입력이 련결되여있는 tty회선에서 이루어진다.
stty는 말단회선동작방식을 변화시키는 설정값아닌 인수로서 다음의것을 받아들인다. 어떤 기능앞에 《-》표식을 한것은 이 기호를 앞에 두어 그 기능을 해제할수 있다는 의미이다. 일부 인수들은 체계에 따라 사용할수 없는 경우도 있다.

su /usr/bin/su

다른 식별이름의 환경을 사용하기

망탈퇴함이 없이 림시로 다른 사용자의 UID, GID를 사용할수 있도록 한다. 주로 고급사용자의 쉘을 사용할 때에 많이 사용된다.

형식 : su[설정값][USER][쉘변수]

-c COMNMAND, —command=COMMAND : 대화형쉘을 시작하지 않고 -c설정값을 쉘에 주어 한개의 지령만을 수행하도록 한다.

-f, —fast : csh와 tcsh에만 해당되는것으로서 -f설정값은 시작파일(.cshrc)을 읽지 않도록 한다.

—help : 표준출력으로 사용법을 출력하고 정상완료한다.

-, -l —login : login스크립트파일을 읽어 사용자의 환경으로 변경한다.

-m, -p, —preserve-environment : 《HOME》, 《USER》, 《LOG NAME》, 《SHELL》 등의 환경변수값을 변경하지 않는다.

-s, —shell shell : su를 실행하는 사용자가 super user가 아니거나 사용자의 쉘이 제한된 쉘일 때에는 /etc/passwd에 등록된 USER의 쉘을 수행하지 않고 지정한 쉘을 수행한다.

—version : 표준출력으로 판본정보를 출력하고 정상완료한다.

su는 한 사용자가 일시적으로 다른 사용자로 될수 있도록 해준다. 실제 사용자 ID, 집단 ID, USER의 보충적인 집단으로 쉘을 실행한다. USER가 주어지지 않으면 기본적으로 고급사용자인 뿌리로 설정된다. 실행되는 쉘은 USER의 암호목록에서 찾아오거나 없으면 /bin/sh를 수행한다.

만약 USER에 암호가 있다면 su는 실제 사용자 ID가 0(고급사용자)이 아닌 한 암호를 물어온다.

기본적으로 su는 현재의 파일등록부를 변경하지 않는다. USER의 암호항목으로부터 HOME, SHELL 등의 변수를 설정하고 만약 고급사용자가 아니라면 USER와 LOGNAME을 USER로 설정한다. 기본적으로 이 쉘은 login쉘이 아니다. 현재 login쉘은 admin이다.

실례를 통해 보기로 하자. 아래와 같이 id명령으로 확인한 결과, 현재 login쉘이 admin이라는것을 알수 있다.

```
kcc@linux:~
$ id
uid=500(admin) gid=500(admin) groups=500(admin)
```

su명령에 의해 뿌리권한쉘로 변경할수 있다.

```
$ su
```

su명령은 뿌리의 PATH는 가져오지 못하므로 su -명령을 사용할것을 권고한다.

```
$ su -
```

또는 다음과 같이 명령을 줄수도 있다.

```
$ su -l
```

```
$ su --login
```

```
kcc@linux:~
# id
uid=0(root) gid=0(root) groups=0(root), 1(bin),2(daemon),
3(sys),4(adm),6(disk),10(wheel)
```

sum

/usr/bin/sum

파일확인

파일의 검사합(checksum)과 블로크수를 계산하여 그것이 원본파일과 꼭 같은가를 확인하는 명령문이다.

형식 : sum[설정값]파일

-r : 설정값을 무효로 하고 BSD sum알고리듬과 1KB블로크를 리용한다.
-s, —sysv : sys V sum알고리듬과 512Byte블로크를 리용한다.
—help : 도움말을 보여준다.
—version : 판본정보를 보여준다.

sum명령은 입력받은 파일을 읽어 해당파일의 1KB블로크수와 검사합을 계산하는 기능을 하며 만일 설정값을 지정하지 않으면 기본값인 BSD sum알고리듬을 사용하게 된다. 이 명령은 일반적으로 통신상에서 주고 받은 파일들간에 그것이 원본파일에 대한 사본파일이며 도중에 변화되지 않았는가를 알아보기 위해 유용하게 리용될수 있다.

아래와 같이 원본파일과 대상파일을 지정하여 그것의 1024MB당 블로크수와 검사합을 확인할수 있다. 이 파일은 원본파일과 같은것으로 판단할수 있다.

```
kcc@linux:~
$ sum  type  type1
10914  8 type
10914  8 type1
```

swapoff /usr/bin/swapoff

swap중지

파일 및 장치에 설정하였던 교환(swapping)을 끝내는 명령문이다.

형식 : swapoff -a[장치명]

-a : /etc/fstab에 swap장치로 표기된 모든 장치를 swap에서 해제한다.

swapoff명령은 파일이나 path로 분류되는 블로크형장치로의 교환(swapping)을 중지한다.

【관련명령문】

mkswap : swap령역을 지정하는 명령문
swapon : swapping을 설정하는 명령문

참 고

swap

교환(swap)이란 주기억장치의 자료를 림시적으로 보관해둘수 있는 하드디스크의 공간으로서 조작체계에서 자료를 필요로 하는 경우에 주기억장치의 내용과 디스크상의 내용을 바꾸는 역할을 한다. 교환(swapping)은 체계의 주기억장치보다 큰 프로그람이나 자료파일을 취급할수 있게 한다.

swapon /usr/bin/swapon

swap설정

파일 및 장치를 교환(swap)으로 설정하는 명령문이다.

형식 : swapon[설정값][장치명]

-a : /etc/fstab에 sw'' swap장치로 표기된 모든 장치를 swap로 설정한다.
-p 우선권 : swapon에서 사용되는 우선권을 설정한다. 우선권은 0~32767 사이의 값이다. pri=value라는 항목을 /etc/fstab의 설정값항목에 써넣으면 swapon -a할 때에 리용된다.

swapon은 파일이나 path로 분류되는 블로크형장치로의 swap령역을 설정한다.

512MB의 swap파일만들기

```
kcc@linux:~
# dd if=/dev/zero of/swap bs=1024 count=524288
# mkswap /swap 524288
# sync
# swapon /swap
```

free명령으로 작성된 교환파일을 확인한다.

```
# free
```

swap를 해제하기 위해서는 swapoff명령을 사용한다.

```
# swapoff /swap
# free
```

sync /usr/bin/sync

주기억기의 내용을 보관

주기억기를 디스크자료로 동기화하여 보관하는 명령문이다.

형식 : sync[--help][--version]

—help : 도움말을 보여준다.

—version : 판본정보를 보여준다.

sync는 모든 디스크밖의 완충화(buffering)된 자료를 기록한다. 이것은 super block나 i-node의 변경사항은 물론 지연된 읽기와 쓰기를 포함할수 있다. 핵심부(kernel)는 비교적 속도가 뜬 디스크에 읽기, 쓰기를 하는 과정에서의 처리속도지연을 피하기 위해 주기억기상에 자료를 보관하는데 이것은 실행속도를 증가시키지만 만일 콤퓨터가 파손되는 경우에는 자료의 류실이나 파일체계의 파괴를 가져올수 있다. sync는 CPU가 비정상으로 정지되기전에 호출되여야 한다. 대체로 CPU는 shutdown이나 reboot halt명령문을 사용하였을 때에 정지된다.

참 고

동기화(synchronization)

동기화는 서로 다른 속도로 동작하는 장치호상간에 원활하게 동작할수 있도록 동작의 진행을 일치시키는 작업을 말한다. UNIX에서는 림시적인 주기억기상의 내용을 하드디스크에 보관하는 경우에 동기화(sync)한다고 말한다.

T

tac /usr/bin/tac

파일내용을 역순으로 출력

파일내용을 마지막행부터 보여준다.

형식 : tac[설정값]파일

-b, -before : 구분하는 식별자를 미리 보여준다. 기본적으로 행바꿈을 표시한다.
-r, -regex : 정규표현식으로 식별자를 사용한다.
-s. -separator=문자렬 : 검색한 문자렬을 기준으로 문장을 자른다.
-help : 도움말을 보여준다.
-version : 판본정보를 보여준다.

tac은 파일내용을 거꿀행순서로 출력한다. 마지막행부터 시작하여 문서(파일)의 앞방향으로 향한(원래의 순서와 반대방향)순서로 보여준다.
먼저 파일내용을 순서대로 출력하는 cat명령을 리용하여 파일을 보자.

```
kcc@linux:~
# cat ics
××정보쎈터의 배구선수명단

1. 김철수
2. 서영식
3. 최명일
4. 정대철
5. 송광식
6. 차철호
```

tac명령은 아래로부터 우의 방향으로 파일을 출력한다.

```
kcc@linux:~

# tac irc
6. 차철호
5. 송광식
4. 정대철
3. 최명일
2. 서영식
1. 김철수

××정보쎈터의 배구선수명단
```

《4.》를 기준으로 잘라서 《4.》아래를 먼저 출력하고 다시 문서의 처음을 보여주려면 다음과 같이 《-s4》로 《4.》라는 문자렬을 검색하여 잘라준다. 그러면 《4.》아래부분이 먼저 출력되고 다시 앞부분부터 《4.》전까지 출력된다.

```
kcc@linux:~

# tac -b   s4 irc
4. 정대철
5. 송광식
6. 차철호
××정보쎈터의 배구선수명단

1. 김철수
2. 서영식
3. 최명일
```

【관련명령문】

cat : 파일을 우로부터 아래방향으로 보여준다.

head : 파일의 첫행부터 10행 또는 설정한 행만큼 보여준다.

tail : 파일의 마지막 행부터 10행 또는 설정한 행만큼 보여준다.

rev : 행마다 본문을 반대로 읽어들인다.

tail

/usr/bin/tail

파일의 뒤부분보기

파일의 뒤부분부터 출력하려는 행까지 지정하여 출력한다.

형식 : tail[설정값]파일

-n[단위] : 파일의 마지막부터 n번째까지 보여준다.
l 행
c 문자
-c num[단위], -byte num[단위] : 마지막 num byte를 보여준다.
b 512bytes
k 1024bytes
-q, -quiet, -silent : 출력전에 파일이름을 보여주지 않는다(기본적으로 보여주지 않는다).
-v, -verbose : 출력전에 파일이름을 보여준다.
-help : 도움말을 보여준다.
-version : 판본정보를 보여준다.

tail명령은 문서의 마지막 행부터 시작하여 지정한 행까지 보여준다. 아무러한 설정도 하지 않는 경우에는 10행을 출력한다. 이 명령은 파일의 앞부분을 출력하는 head명령과 반대되는 기능을 수행한다.

출력하려는 부분에 대한 설정은 기본적으로 행을 사용하지만 설정에 따라 용량, 문자개수로 구분되기도 한다.

보통 파일의 내용을 보여주는 명령문으로는 car, more, less 등이 있는데 tail은 파일의 수많은 내용가운데서 마지막 행만을 보려고 할 때 주로 리용한다. 대표적으로는 최근의 리력파일(log file)을 볼 때에 많이 사용된다.

/var/log/messages리력을 tail명령문으로 보도록 하자. Tail명령문은 파일의 마지막 10행을 보여준다. 알아보기 쉽게 파이프(|)로 nl명령을 입력하였다. nl명령은 행수를 보여주는 명령문이다.

```
kcc@linux:~

# tail /var/log/messages | nl
1 Jun 17 14:18:10 ns cardmgr[545]: no pcmcia driver in /proc/devices
2 Jun 17 14:18:10 ns cardmgr[5451]: exiting
3 Jun 17 14:20:49 ns kernel: VFS: busy inodes on changed media.
4 Jun 17 22:22:11 ns named[13376]: lame server resolving '165.155.49.
2211.in-addr.arpa' (in '155.49.211.in-addr.arpa'?): 210.117.65.100#53
5 Jun 17 22:22:11 ns named[13376]: lame server resolving '165.155.49.
2211.in-addr.arpa' (in '155.49.211.in-addr.arpa'?): 210.117.65.2#53
6 Jun 17 22:22:11 ns named[13376]: lame server resolving '165.155.49.
2211.in-addr.arpa' (in '155.49.211.in-addr.arpa'?): 210.117.65.100#53
7 Jun 17 22:22:11 ns named[13376]: lame server resolving '165.155.49.
2211.in-addr.arpa' (in '155.49.211.in-addr.arpa'?): 210.117.65.2#53
8 Jun 17 22:22:11 ns sshd[6114]: Could not reverse map address 211.
49.155.165.
9  Jun 17 22:22:14 ns  sshd[6114]: Accepted password for root from
211.49.155.165 port 1058 ssh2
10  Jun 17 22:22:14 ns PAM_pwdb[6114] (sshd) session opened for u
ser root by(uid=0)
```

-5설정값은 마지막 행부터 5행을 보여준다. 우의 6번째부터 10번째까지의 내용을 출력하도록 하는 실례를 아래에 보여준다.

```
kcc@linux:~

# tail -5 /var/log/messages | nl
1 Jun 17 22:22:11 ns named[13376]: lame server resolving '165.155.4
9.2211.in-addr.arpa'(in '155.49.211.in-addr.arpa'?): 210.117.65.100#53
2 Jun 17 22:22:11 ns named[13376]: lame server resolving '165.155.4
9.2211.in-addr.arpa' (in '155.49.211.in-addr.arpa'?): 210.117.65.2#53
3 Jun 17 22:22:11 ns sshd[6114]: Could not reverse map address 211
.49.155.165.
4 Jun 17 22:22:14 ns sshd[6114]: Accepted password for root from 2
11.49.155.165 port 1058 ssh2
5 Jun 17 22:22:14 ns PAM_pwdb[6114] (sshd) session opened for us-
er root by(uid=0)
```

wc명령으로 messages리력의 행수는 53501이라는것을 확인할수 있다.

```
kcc@linux:~
# wc /var/log/messages
53501 815088  5839075 /var/log/messages
```

53501의 행에서 53000행부터 나머지의 마지막까지 출력하도록 하려는 경우에는 +53000설정값을 사용한다. 《+행수》는 처음부터 제거하려는 행까지 지정하는 기능을 한다.

```
# tail +53000/var/log/messages
```

【관련명령문】

cat : 파일을 첫 행부터 아래방향으로 보여준다.
tac : 파일을 마지막 행부터 우의 방향으로 보여준다.
head : 파일의 첫행부터 10행 또는 설정한 행만큼 보여준다.

talk

/usr/bin/talk

1 대 1 의 대화

말단에 접속한 사용자와 대화를 나눌수 있게 하는 명령문이다.

형식 : talk[사용자ID][tty이름]

사용자ID : talk로 접속할 식별이름
tty이름 : 같은 식별이름이라면 tty이름을 지정해준다. tty** 또는 pts/x로 지정해준다. 현재 체계에 접속중인 식별확인에는 《finger》지령을 리용한다.

talk는 말단에서 사용하는 실시간대화프로그람으로 생각하면 된다. 같은 체계에 있는 사용자와의 대화나 다른 체계사용자와의 대화가 모두 가능하다.
talk를 사용하기 위해서는 talk봉사데몬이 동작하고있어야 한다.

talk로 같은 체계에 접속한 다른 ID의 사용자와 접속하려면 《talk [사용자ID]》의 서식으로 사용한다. 만약 같은 ID로 접속한 다른 말단의 사용자와 대화하려는 경우에는 아래와 같이 tty이름을 붙여준다.

```
kcc@linux:~
# talk  pirania pts/2
[Waiting for your party to respond]
```

대화하려는 사용자말단을 지정하면 상대방의 응답을 기다리게 된다.

상대방의 창에는 다음의 통보문이 현시되면서 대화를 요청하고있다는것을 알려준다. 상대방도 우에서 한것과 같이 대방에게 접속해주어야 한다.

```
kcc@linux:~
Message from Talk_Daemon@localhost.localdomain at 0:24...
talk: connection  requested by pirania@localhos.localdomain
talk: respond with: talk pirania@localhost.localdomain
# talk pirania
```

서로 talk로 접속하면 창이 우와 아래의 두개로 나누어져 우에는 자기가

입력하는 내용이 아래에는 상대방이 입력한 내용이 실시간으로 나타난다.

현재 가입하지 않은 다른 체계인 command.kccbook.co.kp에 first라는 ID와 대화하려는 경우에는 다음과 같이 접속한다.

```
# talk first@command.kccbook.co.kp
```

talk를 사용하려면 talk봉사기데몬이 시작되여있어야 하지만 일반적인 Linux에서는 talk봉사기로 ntalk를 사용한다.

참 고

ntalk(New Talk)

talk를 사용하려면 talk봉사기가 활성화되여있어야 한다. talk는 Sun회사제품의 조작체계에서 사용되던것으로써 현재 거의 모든 체계에 ntalk가 설치되여있다.

실례로 ntalk는 RedHat Linux체계인 경우에 xinitd데몬에 포함되여있다(xin - itd데몬설정을 참고할것). talk와 ntalk는 호상간에 517포구와 518포구로 호환되지 않는다.

IRC(Internet Relay Chat)

IRC는 일종의 실시간대화체계로서 의뢰기 및 봉사기구조의 쏘프트웨어이다. IRC 사용자는 실시간대화집단(Charting Group)을 새로 만들거나 아니면 이미 있던 실시간대화집단에 합류할수 있다. 이미 있던 실시간대화집단과 그에 속한 성원들을 찾아볼수 있게 하는 통신규약도 있다.

망의 종류에 따라서는 자기를 가리키는 별명을 등록하여 대화진행동안 계속 사용할수 있다. 일부 실시간대화마당에서는 항상 사용할수 있는 별명을 등록하도록 유도하며 개인프로파일(profile)이나 사진 및 개인홈페지를 련결(link)할수 있는 공간을 제공하는것도 있다. IRC규약은 TCP를 사용하며 대체로 6667번포구(Port 6667)를 사용한다.

tar

/usr/bin/tar

파일묶기

여러개의 파일과 등록부를 하나의 파일로 묶는다.

형식 : tar[설정값]파일이름1 파일이름2

파일명1 : 파일을 압축 또는 묶움으로 만들어진 결과물로 나올 파일명을 지정한다.

파일명2 : 압축이나 묶음으로 만들 파일들을 지정한다.

-x, —extract, —get : 문서고(archive)에서 파일을 푼다.

-c, —create : 새로운 문서고를 만든다.

-v, —verbose : 처리되는 내용의 상세한 정보를 보여준다.

-z —gzip, —ungzip : gzip로 압축한다. ungzip로 압축을 해제한다.

-j -l —bzip : bzip2를 리용하여 압축한다.

-t, —list : 압축되는 목록을 보여준다.

-d, —diff, —compare : 문서고와 파일체계간의 차이점을 비교한다.

-f, —file=ARCHIVE : 설정값다음의 이름지정으로 기본값으로 설정된 이름을 변경할수 있다.

-r, —append : 문서고끝에 새로운 파일을 추가적으로 생성한다.

-u, —update : 문서고안의것보다 새로운 파일만을 추가한다.

-A, —catenate : 문서고에 tar파일을 추가한다.

-w, —interactive : 모든 행동에 대해 확인을 요구한다.

—help : 도움말을 보여준다.

—version : 판본정보를 보여준다.

tar는 파일이나 파일등록부를 하나의 파일로 묶는 역할을 한다. 파일의 예비보관이나 전송때에 하나의 파일로 결합하는데 주로 사용된다. 파일을 압축하는것으로는 gzip나 bzip2이 많이 사용된다.

```
kcc@linux:~

# pwd
/var/www
# ls
cgi-bin html icons nut-cgi-bin test_html workdesk www_html
```

파일묶기

WWW_html파일등록부를 backup.tar파일로 묶어보자. tar로 파일을 묶는 경우에 대체로 -cf설정값을 많이 사용하게 된다. c는 새로 tar파일을 만드는 역할을 하며 f는 문서고명을 지정한다.

```
kcc@linux:~

# tar -cf backup.tar WWW_html/
# ls -alh
합계 104MB
dtwxr-xr-x   9 admin  admin  4.0K  6월 14  09:43 .
drwxr-xr-x  27 root   root   4.0K  3월 5 17:00 ..
-rw-r—r—     1 root   root   103M  6월 14 09:43 backup.tar
drwxr-xr-x   2 admin  admin  4.0K  2월 1 15:46 cgi-bin
drwxr-xr-x   5 admin  admin  4.0K  2월 4  15:06 html
drwxr-xr-x   3 admin  admin  4.0K  2월 4 15:06 icons
dtwxr-xr-x   2 admin  admin  4.0K  12월 14 19:58 nut-cgi-bin
drwxr-xr-x   8 admin  admin  4.0K  6월 11  16:54 test_html
drwxr-xr-x   2 root   root   4.0K  5월 2  15:11 workdesk
drwxr-xr-x   3 admin  admin  4.0K  5월 20 11:48 www_html
```

파일보기

-tvf설정값을 사용하면 tar로 묶인 파일을 실지로 푸는것이 아니라 파일의 내용을 볼수 있게 한다.

```
kcc@linux:~

# tar -tvf backup.tar  |  more
drwxr-xr-x admin/admin  0 2002-06-11 16:54:33 www_html/
-rw-r — r —   admin/admin  198727  2001-12-14  21:09:33  www_html/
proftpd_conf.html
```

```
drwxr-xr-x admin/admin      0 2002-04-04 17:06:23 www_html_howto/
-rw-r — r —     admin/admin     20188   2002-04-04   17:06:23
www_html/howto/test.htm
-rw-r — r —     admin/admin    13162   2002-03-29   13:19:46
www_html/howto/001.GIF
-rw-r — r —     admin/admin     6880   2002-03-29   13:19:46
www_html/howto/002.GIF
    ————————————— 중간 생략 —————————————
-rw-r — r —     admin/admin      8328   2002-03-29   13:19:46
www_html/howto/014.GIF
-rw-r — r —     admin/admin      8405   2002  -03-29  13:19:46
www_html/howto/015.GIF
-rw-r — r —     admin/admin    14974     2002-03-29   13:19:46
www_html/howto/016.GIF
-rw-r — r —     admin/admin     9703     2002  -03-29  13:t9:46
www_html/howto/017.GIF
-rw-r — r —     admin/admin    13551     2002-03-29   13:t9:46
www_html/howto/018.GIF
```

묶음풀기

-xf설정값을 사용하면 backup.tar파일의 묶음을 풀게 된다. 한편 -xvf설정값을 사용하면 파일이 처리되는 내용을 상세히 보여준다.

```
# tar -xf backup.tar
```

묶음을 압축하기

그러면 다시 www_html파일등록부의 내용을 묶는것과 동시에 gzip로 압축해보자. -cf설정값에 gzip의 z설정값을 함께 추가하면, 즉 czf설정값을 사용하면 압축까지 할수 있다. 파일이름은 tar파일과의 구별을 위해 일반적으로 tar.gz나 tgz의 확장자를 사용한다. 만일 -v설정값까지 추가하면 명령의 실행처리과정을 볼수 있게 된다.

```
kcc@linux:~
# tar -czf backup.tar.gz www_html/
# ls -alh
합계 840K
drwxr-xr-x  9 admin  admin  4.0K  6월  17 23:10 .
drwxr-xr-x  27 root  root   4.0K  3월  5  17:00 ..
-rw-r—r—    1 root   root   00K   6월  17 23:11 backup.tar.gz
drwxr-xr-x  2 admin  admin  4.0K  2월  1 15:46 cgi-bin
drwxr-xr-x  5 admin  admin  4.0K  2월  4 15:06 html
drwxr-xr-x  3 admin  admin  4.0K  2월  4 15:06 icons
drwxr-xr-x  2 admin  admin  4.0K  12월 14 2001 nut-cgi-bin
drwxr-xr-x  8 admin  admin  4.0K  6월  11 16:54 test_html
drwxr-xr-x  2 root   root   4.0K  5월  2  15:11 workdesk
drwxr-xr-x  admin    admin  4.0K  5월  20 11:48  www_html
```

-xzf설정값으로 backup.tar.gz파일을 압축해제할수 있다. 물론 gzip -d 설정값으로 gzip압축을 해제한 다음 tar-xf설정값으로 tar묶음해제를 할수도 있다.

```
# tar-xzf backup.tar.gz
```

묶여진 tar파일은 u설정값을 사용하여 갱신할수 있다. 즉 원본파일이 변경되거나 파일목록을 추가할 때마다 tar파일을 새로 묶지 않고 변화된 파일만 갱신할수 있는것이다. 첫번째 실례로 생성된 backup.tar를 갱신해보자.

```
kcc@linux:~
# mkdir /var/www/www_html/image
#  tar -uvf backup.tar www_html/
image/
```

image/파일등록부가 backup.tar파일에 갱신된것을 보여준다.

tee /usr/bin/tee

같은 내용을 두 경로로 동시에 출력

입력한 내용을 화면에 보여주는것과 동시에 파일로 보관한다.

형식 : tee[설정값]파일

파일 : 보관되는 파일명
-a, -append : 덧쓰기하지 않고 주어진 파일에 표준입력을 추가한다.
-i, -ignore-interrupts : 새치기신호를 무시한다.
-version : 판본정보를 보여준다.
-help : 도움말을 보여준다.

tee명령은 입력된 내용을 파일에 추가하는데 사용된다.

같은 이름의 파일이 이미 있는 경우 tee명령만 사용하면 파일을 덧쓰기하게 되며 -a설정값을 사용하면 기존파일에 입력된 내용이 추가된다.

명령이나 파이프(|)를 통해 나오는 출력을 받아 화면에 보여주거나 파일에 보관되는데 출력내용에 대하여서는 아무러한 영향도 미치지 않는다.

```
kcc@linux:~
# cat  maillist
piranla@empal.com
cyc-x-1@kccmail.net
moolli@dreamwiz.com

# cat maillist  |  sort  |  tee maillist_sort
cyc-x-1@kccmail.net
moolli@dreamwiz.com
pirania@empal.com
```

maillist내용을 파이프로 받아 sort명령으로 정렬하고 그 결과를 tee를 통해 화면으로 출력하는것과 동시에 maillist_sort로 보관한다.

telinit

/usr/bin/telinit

프로세스조종

특정준위의 체계구성으로 설정한다.

형식 : telinit runlevel

telent는 현재 준위의 모든 프로세스를 완료하고 설정한 준위의 프로세스를 시작한다. /etc/inittab파일의 설정에 따라 실행준위를 수행한다.

X-window체계는 보통 runlevel 5로 되여있다. 이 상태에서 xdum을 모두 완료하려면 telinit를 리용하여 runlevel 3으로 만들어준다.

```
# telinit 3
```

체계완료에 관계된 runlevel 6을 지정하면 체계는 완료된다.

```
# telinit 6
```

telnet

/usr/bin/telnet

원격접속

telnet를 통하여 호스트에 원격접속한다.

형식 : telnet[설정값][호스트[포구]]

호스트 : 접속할 호스트로서 인터네트주소형식으로 사용된다.
포구 : 접속에 리용할 호스트의 포구이다. 지정되지 않으면 telnet에서의 초기값이 사용된다(23번포구).
-l 사용자ID : 사용자ID telnet봉사기체계에 접속할 식별이름을 지정한다.
-a : 현재 사용자이름을 ID 로 사용하여 접속한다.

telnet는 원격말단을 지원하는 의뢰기이다. 정확히 말하면 원격체계를 지원하기 위한 명령문과 TCP/IP토대의 통신규약이다.

telnet는 접속하려는 봉사기의 IP나 령역으로 접속가능하다.

```
kcc@linux:~

# telnet   www.linuxroot.co.kp
Trying 168.126.61.21...
Connected to www.linuxroot.co.kp(168.126.61.21)
Escape  character Is   '^]'.

kcccomLinux release 2.2
Kernel 2.4.13-1hl on an i1686
login:
```

USER ID를 지정하면 접속시에 따로 login을 묻지 않고 해당 ID의 암호입력으로 접속할수 있다. 《-a》설정값을 사용하면 현재 사용중인 식별이름으로 접속한다.

```
# telnet -l first www.linuxroot.co.kp
```

접속시도할 체계의 포구를 지정할수 있다. telnet봉사는 일반적으로 23번을 사용한다. 그러나 봉사기보안이나 관리자의 특별한 리유로 인하여 telnet의 봉사포구를 변경하는 경우가 있다. 이러한 경우에 접속포구를 지정해준다.

```
kcc@linux:~
# telnet www.linuxroot.co.kp  5050
Connected to www.linuxroot.co.kp(168.126.61.21)
Escape character is  '^]'.

kcccomLinux release 2.2
Kernel 2.4.13-1hl on an i686
login:
```

봉사하는 포구에 직접 접속하여 봉사여부를 알수 있다. 웨브봉사여부를 알아보자. Connected to통보문에 의해 웨브봉사포구(80번)가 열린것을 알수 있다.

```
kcc@linux:~
# telnet www.linuxroot.co.kp 80
Trying  168.126.61.21...
Connected to www.linuxroot.co.kp(168.126.61.21)
Escape character  is '^]'.
```

아래에서는 FTP포구로 접속을 시도해본다. 〈Connection refused〉통보문을 통하여 봉사가 되지 않는다는것을 알수 있다.

```
kcc@linux:~
# telnet www.linuxroot.co.kp 80
Trying  168.126.61.21...
telnet: Unable to connected to remote host: Connection re fused
```

telnet사용명령문

접속중에도 [ctrl+]건입력에 의해 명령방식으로 설정할수 있다.

명령문	설 명
logout	사용자가 logout하여 접속을 끊는다.
display	인수를 출력한다.
mode	문장이나 문자의 방식을 변경한다.《mode ?》로 도움말을 볼수 있다.
open	호스트나 IP를 지정하여 열려는 싸이트나 호스트를 정한다.
quit	TELNET를 완료한다
send	특수문자를 보낸다. 《send ?》로 도움말을 볼수 있다.
set	인수를 설정한다. 《set ?》로 도움말을 볼수 있다.
unset	인수설정을 해제한다. 《unset ?》로 도움말을 볼수 있다.
status	telnet상태를 출력한다.
toggle	인수를 고정한다. 《toggle ?》로 도움말을 볼수 있다.
slc	특수문자의 상태를 변경한다. 《slc ?》로 도움말을 볼수 있다.
z	telnet를 정지한다.
!	쉘명령문을 사용할수 있다.
environ	환경변수를 변경한다. 《environ ?》로 도움말을 볼수 있다.
?	도움말을 보여준다.

test

/usr/bin/test

검사

파일류형을 검사하고 값을 비교하는 명령문이다.

형식 : test[표현식]
test[설정값]

—help : 사용법을 보여준다.
—version : 판본정보를 보여준다.

조건문에서 사용되여 다음에 나오는 파일이나 문자렬의 인수를 검사한다. 참인 경우에 0값을 반환하며 거짓인 경우에는 0이 아닌값을 반환한다.

조건문에서 편리하게 사용할수 있다. test명령이 어떤 명령을 수행하기보다는 조건문에 사용되여 파일이나 문자렬을 판단하는데 사용되므로 다음과 같은 실례를 통해 보는것이 빨리 리해할수 있게 한다.

```
kcc@linux:~
# if test -e "mbox"; then echo "참" ; fi
참
```

간단한 if조건문이다. if조건문을 모르는 경우에 shell부분을 참고한다.

우의 실례에서 《-e》는 표현식으로 파일의 존재여부를 확인한다. 파일이 있으면 참값을 반환한다. 여기서는 《mbox》라는 파일이 존재하면 then이후에 《참》을 보여주며 만약 존재하지 않으면 그냥 완료된다. 《참》을 출력하였으므로 《mbox》파일이 있음을 확인할수 있다. 확인하려는 파일이나 문자렬에 맞게 표현식을 적용하여 검사한다.

표현식

표 현 식	설 명
-b	파일이 블로크특수파일이면 참.
-c	파일이 문자형장치파일이면 참.
-d	파일등록부이면 참.
-e	파일이면 참
-f	파일이 보통 파일이면 참
-g	파일이 set-group-id파일이면 참
-k	파일이 《sticky》비트설정의 파일이면 참
-L	파일이 기호련결파일이면 참
-p	파일이 파이프파일이면 참
-r	파일이 읽기가능한 파일이면 참
-s	파일이 0보다 큰 크기를 가지는 파일이면 참
-S	파일이 존재하고 소케트이면 참
-t	fd가 말단상에서 열어진것이면 참. fd가 생략되면 기본값은 1(표준출력).
-u	파일이 존재하고 set-user-id비트설정을 가지면 참
-w	파일이 쓰기가능한 파일이면 참
-x	파일이 실행가능한 파일이면 참
-O	파일이 존재하고 유효사용자ID의 소유이면 참.
-G	파일이 존재하고 유효집단ID의 소유이면 참.
파일1 -nt 파일2	파일1이 파일2보다 최근에 생성되였거나 수정되였다면 참
파일1 -ot 파일2	파일1이 파일2보다 오래된것이면 참
파일1 -ef 파일2	파일1과 파일2가 같은 장치, inode번호를 가진다면 참
-z	문자렬의 길이가 0이면 참
-n	문자렬의 길이가 0이 아니라면 참
문자렬1=문자렬2	두 문자렬이 같으면 참
문자렬1 !=문자렬2	두 문자렬이 같지 않으면 참
! 표현식	표현식이 거짓이면 참
표현식1 -a 표현식2	표현식1과 표현식2가 둘다 참이면 참
표현식1 -o 표현식2	표현식1 또는 표현식2의 둘중 어느 하나라도 참이면 참

참 고

핵심부(kernel)와 Linux

핵심부는 조작체계의 핵심부분을 의미하며 프로그람이 실행될 때에 체계를 리용할수 있도록 중개하는 역할을 담당한다. 우리가 흔히 말하는 Linux는 핵심부와 체계프로그람, 패키쥐화된 일반응용프로그람의 모두를 포함하여 이르는것이다. 이러한 일반응용프로그람들은 많은 경우에 GNU프로젝트에 의해 개발되고 개선발전되였다.

모든 조작체계들에 핵심부가 들어있지만 특히 Linux를 론할 때에 핵심부를 빼놓지 않는것은 핵심부를 직접 번역하여 매 사용자들이 자신에게 적합하게 설정할수 있고 혹은 자체의 오유수정프로그람을 추가해넣을수도 있을 정도로 전용의 핵심부를 각자가 만들어 사용할수 있기때문이다. 핵심부는 일반적으로 /파일등록부나 /boot 파일등록부에 ZImage라는 이름으로 압축되여 생성된다.

tftp

/usr/bin/tftp

tftp봉사기접속의뢰기프로그람

tftp봉사의뢰기(Trivial File Transfer Protocol)

형식 : tftp[호스트]

TFTP는 FTP보다 간단한 기능의 TFTP(Trivial File Transfer Protocol)봉사를 하는 호스트접속의뢰기프로그람이다. 사용자인증이 필요없으며 파일등록부구조를 보여주지 않는다. 보통의 사용법은 ftp와 같다.

원격호스트접속은 IP나 령역이름을 지정하면 된다.

```
# tftp linuxroot.co.kp
```

Tftp사용명령문

명 령 문	설 명
?	사용법에 관한 정보를 보여준다.
acsii	mode ASCII와 같다.
binary	mode binary와 같다.
status	현재 tftp의 상태를 보여준다.
verbose	tftp상태를 자세히 보여준다.
mode	전송방식(ASCII or binary)으로 설정한다.
connect 호스트[포구]	지정한 호스트와 포구로 접속시도한다.
get 파일이름	원격접속호스트의 파일을 국부체계에 내리적재한다.
get 파일이름	국부파일이름 원격접속호스트의 파일을 지정한 국부파일이름으로 내리적재한다.
put 파일이름	국부파일을 원격접속호스트에 올리적재한다.
put 국부파일이름 원격파일이름	국부파일이름을 원격접속호스트에 지정한 원격파일이름으로 올리적재한다.
quit	tftp접속을 끝낸다.

참 고

사용자 public_html파일등록부를 자동적으로 만들기

보통 식별이름에서 웨브봉사를 하는 경우 home등록부의 public_html파일등록부 아래에 index.html파일을 생성하게 된다. 아파치설정파일인 httpd.conf의 UserDir지시자에서 지정한 파일등록부를 생성하게 되는것이다. 기본값으로 UserDir public_html이 설정되여있다.

식별이름을 추가할 때마다 자동적으로 public_html파일등록부를 만들게 하려면 /etc/skel파일등록부아래에 public_html파일등록부를 생성해두면 된다.

```
kcc@linux:~

# ls -al /etc/skel/public_html
drwxr-xr-x   4 root   root    4096 8월 22  2001 public_html
```

top

/usr/bin/top

프로쎄스상태 보여주기

체계의 프로쎄스 및 주기억기의 현재 사용정형을 실시간적으로 보여준다.

형식 : top[설정값]

-b : 일괄방식으로 정보를 출력한다. 실시간의 호상정보를 계속적으로 화면에 일렬로 나타내게 된다.
-d delay : 지정한 delay시간(초)만큼 정보를 갱신하여 보여준다.
-i idle : 프로쎄스나 zombie프로쎄스를 보여주지 않는다.
-n num : 지정한 수자만큼 갱신정보를 보여준 후에 완료한다.
-p pid : 지정한 프로쎄스ID의 정보만을 보여준다.
-q : 시간지연없이 계속된 갱신정보를 보여준다
-s : 일부 대화식지령을 비활성화한다(Secure방식)
-S : 루적자료를 보여준다(cumulative 방식).

top는 체계의 프로쎄스 및 주기억기의 사용상태를 실시간적으로 보여주는 프로그람이다.

top명령은 매 5초마다 체계의 프로쎄스의 상태를 갱신하여 보여준다. 현재 시간, 체계갱신시간, 체계의 망가입사용자수, 지난 1분, 5분, 15분간의 체계평균부하를 보여준다. 아래행에는 프로쎄스정보, CPU상태, 주기억기와 교환(swap)상태를 보여준다.

top명령으로 실시간프로쎄스정보를 보는 상태에서 실행가능한 명령문은 h명령으로 볼수 있다.

Top설정명령문

명 령 문	설 명
space	정보를 갱신한다.
^L	출력화면을 다시 갱신한다
F or f	항목을 추가하거나 제거한다(아래의 《top항목》을 참고할것). 정보보기를 원하는 항목의 자모를 누르면 선택/취소된다.

명 령 문	설 명
r	프로세스의 우선권순위를 변경한다.
N	pid정보로 정렬한다.
A	age정보로 정렬한다.
P	CPU사용량으로 정렬한다.
M	적재된 주기억기사용량으로 정렬한다
T	시간 및 루적시간으로 정렬한다.
u	지정한 사용자관련정보만을 보여준다.
O or o	보여주는 항목의 정렬순서를 바꾼다(아래의 《top항목 》을 참고할것). 순서를 설정한다. 대문자로 선택한 항목을 누르면 왼쪽으로 이동하고 소문자를 누르면 오른쪽으로 이동한다.
h or ?	명령문을 보여준다.
k	프로세스를 제거한다.
n or #	보여주는 프로세스의 수를 지정한다.
s	보여주는 정보의 갱신시간을 정한다.
W	~/.toprc에 설정된 내용을 보관한다.
q	top를 빠져나온다.

top보기의 수정

기본창(main window)의 상단에 보여주는 정보를 수정하는데 쓰인다.

명령문	설 명
S	cumulative mode(실시간정보를 루적자료로 보여줌)를 선택해제한다.
i	idle프로세스정보를 보여주거나 해제한다.
I	Irix나 Soiaris정보를 보여주거나 해제한다.
c	명령행에서 실행된 명령자체로 보여주거나 해제한다.
l	평균부하정보를 보여주거나 해제한다.
m	주기억기정보를 보여주거나 해제한다.
t	요약된 정보만을 보여주거나 해제한다.

top 정렬

기본창에서 실행하는 명령으로서 현재 정보를 사용자의 요구대로 정렬한다.

```
kcc@linux:~

# top
11:58pm up 1 day, 14:00, 2 users, load average: 0.00 0.00 0.00
82 processes: 81 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 0.0% user, 1.1% system, 0.0% nice, 98.8% idle
Mem: 126040K av, 110600K used, 15440K free, 284K shrd, 1812K buff
Swap: 530064K av, 76732K used, 453332K free 33620K cached
PID USER   PRI NI SIZE RSS SHARE STAT %CPU %MEM  TIME COMMAND
5196 root  18  0  1152  1152  888 R    0.9  0.9  0:00 top
5144 root  10  0   756  1724 1400 S    0.1  1.3  0:00 sshd
1 root     8   0   168   124  124 S    0.0  0.0  0:03 init
2 root     9   0   0     0    0 SW     0.0  0.0  0:00 keventd
3 root     9   0   0     0    0 SW     0.0  0.0  0:00 kapm-idled
4 root    19  19   0     0    0 SWN    0.0  0.0  0:00 ksoftirqd_CPU0
5 root     0   0   0     0    0 SW     0.0  0.0  0:08 kswapd
6 root     9   0   0     0    0 SW     0.0  0.0  0:00 kreclaimd
7 root     9   0   0     0    0 SW     0.0  0.0  0:00 bdflush
8 root     9   0   0     0    0 SW     0.0  0.0  0:00 kupdated
9 root    -1 -20   0     0    0 SW<    0.0  0.0  0:00 drecoveryd
63 root    9   0   0     0    0 SW     0.0  0.0  0:00 khubd
162 root   9   0   0     0    0 SW     0.0  0.0  0:00 kjournald
163 root   9   0   0     0    0 SW     0.0  0.0  0:00 kjournald
709 root   9   0   12    256  256 S    0.0  0.2  0:00 syslogd
714 root   9   0   760   84   84 S     0.0  0.0  0:00 klogd
760 named 9    0   1952  1720 1260 S   0.0  1.30:00  named
```

top항목

항목의 구성을 변경할 때에 나오는 내용을 정리하여 아래에 보여주었다.

부 호	기 호	명 칭	설 명
*	A	PID	Process ID
	B	PPID	Parent Process ID
	C	UID	사용자 ID(User ID)
*	D	USER	사용자이름
*	E	%CPU	CPU사용량
*	F	%MEM	주기억기의 사용량
	G	TTY	사용중인 tty를 보여준다.
*	H	PRI	우선권순위

*	I	NI	Nice Value
	J	PAGEN	페지오유수(Page Fault Numder)
	K	TSIZE	코드크기(kb)
	L	DSIZE	data+stack의 크기(kb)
*	M	SIZE	가상image의 크기(kb)
	N	TRS	현재 문자의 크기(kb)
	O	SWAP	swap크기(kb)
*	P	SHARE	분할된 페지(kb)
	Q	A	접근한 페지수(kb)
	R	WP	쓰기보호된 페지수(kb)
	S	D	쓰레기페지
*	T	RSS	현재 페지가 상주하고있는 크기
	U	WCHAN	함수중 휴식상태에 있는것
*	V	STAT	프로쎄스상태
*	W	TIME	CPU시간
*	X	COMMAND	명령문
	Y	LC	마지막에 사용된 CPU
	Z	FLAGS	검사기발(Task Flag)

[주의]

*는 현재 사용하고있다는 뜻이다.

touch

/bin/touch

빈 파일을 만들기

파일을 만들며 파일에 접근하거나 수정한 시간을 변경한다.

형식 : touch[설정값]파일이름

-a, —time=atime, —time=access,—time=use : 최근파일사용시간(access time)만을 갱신한다.
-c, —no—create : 지정한 파일이 없는 경우에도 파일을 생성하지 않는다.
-d time, —date time : 현재시간대신에 지정한 시간으로 변경한다. (월이름, 시간대, am, pm)
-m, —time=mtime, —time=modify : 파일이 수정된 시간을 변경한다.
-r file, —reference file : 지정한 파일의 시간으로 변경한다.
-t time : 현재시간대신에 지정한 시간으로 변경한다.
([[cc]yy]mmddhhmm[.ss]) [cc]yy:년, mm:달, dd:일. hh:시.
mm:분, ss:초
—help : 도움말을 보여준다
—version : 판본정보를 보여준다.

touch명령은 주어진 파일의 최근 사용시간과 최근 변경시간(파일내용이 바뀐 시간)을 체계의 현재시간으로 변경한다. 만일 파일이 없는 경우에는 0byte크기의 이름만 있는 파일, 즉 빈 파일을 만들게 된다.

이 명령은 파일에 대한 접근 및 수정시간을 현재시간이나 특정한 시간으로 변경시키는 편의프로그람이지만 새로운 파일을 만들 때에 사용할수도 있다.

script.sh라는 파일이 없으면 touch script.sh라는 명령문으로 새로운 파일을 만들게 된다. 이러한 형태로 시간정보를 가지는 빈 파일을 만들어 프로젝트의 변경날자를 추적하는것과 같은데 적용할수 있다.

touch는 파일이 존재하는 경우에 파일의 접근과 수정시간을 변경한다.

시간지정이 없는 경우에는 체계의 현재시간이 적용된다.

```
kcc@linux:~

#  ls -al script.sh
-tw-r—r—     1 root  root   269   5월 12 00:56 script.sh

#  touch script.sh
#  ls -al  script.sh
-rw-r—r— 1 root   root   269   7 월 11 10:40  script.sh
```

파일이 없는 경우 touch명령문을 리용하면 새로운 파일이 만들어지지만 -c설정값을 사용하면 새로운 파일을 만들지 않는다. 또한 -t설정값을 리용하면 새로 만드는 파일의 생성시간을 현재시간이 아닌 시간으로 설정할수 있다.

2004년 6월 6일 00시 00분으로 파일의 수정날자를 변경해보자.

YYMMDDHHMM(YY:두자리수의 년도, MM:월, DD:일, HH:시간, MM:분)형식으로 입력한다.

```
kcc@linux:~

# ls -al script.sh
-rw-r—r— 1 root   root   269   7월 11 10:40 scrlpt.sh
#>touch -c -t  0206060000 script.sh
#>ls-al script.sh
-rw-r—r— 1 root   root   269    6 월 6 00:00 script.sh
```

참　　고

자유쏘프트웨어재단(Free Software Foundation)

략칭은 FSF이다. 리챠드 스톨맨에 의하여 콤퓨터프로그람의 복제, 배포, 수정에 대한 제한들을 철페하기 위한 목적으로 설립되였다. 여기에서의 자유란 어디에도 구속되지 않는다는것, 즉 이전시기의 상업적목적을 추구하던 쏘프트웨어제품에서와는 달리 복제, 배포, 수정의 자유를 의미한다.

tr

/usr/bin/tr

문자의 변환 및 삭제

문자렬에서 특정문자를 삭제하거나 변환한다.

형식 : tr[설정값]문자렬1[문자렬2]

-c, —complement : 문자렬1을 ASCII값 001~337과 비교하여 보수연산을 한다.

-d, —delete : 문자렬1에서 지정한 문자를 삭제하여 보여준다.

-s. —squeeze-repeats : 문자렬2에서 반복되는 문자를 지운다.

-t, —truncate-set1 : 문자렬1을 문자렬2의 길이로 자른다.

—help : 도움말을 보여준다.

—version : 판본정보를 보여준다.

tr는 문자변환 려과기(filter)로서《 》부호나 []부호를 적절하게 사용하여 표준출력으로 보여주는 역할을 한다.

아래의 실례에서는 입력받은 filename의 문자렬가운데서 모든 대문자를 소문자로 수정한 내용을 보여준다.

```
kcc@linux:~

# cat smallletter
abcdefghijklmnopqrstuvwxyz    1234567890

# tr "a-z" "A-Z" < smallletter
ABCDEFGHIJKLMNOPQRSTUVWXYZ    1234567890
```

-d설정값은 입력받은 파일에서 지정한 문자렬을 모두 삭제하여 보여준다.

다음의 실례는 filename에 들어있는 모든 수자들을 삭제한 내용을 보여준다.

```
kcc@linux:~

# tr -d "0-9" < smallletter
abcdefghijklmnopqrstuvwxyz
```

tr는 반복되는 문자를 정리할수 있다. 이것은 -s설정값을 리용하여 하게 되는데 공백(" ")을 지정해주어 련속된 공백을 삭제하고 삭제한 공백을 행바꿈표시인《"\012"》로 바꾸어준다(ASCII코드 012는 행바꿈(newlines)값이다).

```
kcc@linux:~
# tr -s " " "\012" < test
abcdefghijklmnopqrstuvwxyz
1234567890
```

참 고

파일등록부의 사용량보기

파일등록부의 크기를 나타내는 경우 대체로 du명령을 사용한다. 여기에 -s설정값을 같이 사용하면 지정한 파일등록부의 아래준위파일등록부크기까지의 총사용량을 보여준다.

```
kcc@linux:~
# du -sh /home
1.2G    /home
1.2G    합계
```

GPL

Gnu General Public License의 략자로서 FSF에서 GNU사상에 립각하여 공개원천(open source)프로그람에 적용시킨 사용허가권(License)으로서 이전의 사용허가권이 쏘프트웨어의 수정, 공유의 자유를 제한하고있는것에 대처하여 그것을 보장하기 위한 목적으로 만들어져 적용되는것이다.

traceroute

/usr/bin/traceroute

파케트이동경로보기

지정한 호스트에로의 망경로를 보여주는 명령문이다.

형식 : traceroute[설정값]호스트명[파케트크기]

-m : 홉(hop)을 지정한다.
(기본값은 30이고 최대값 255까지 지정가능)
-n : 주소찾기를 끌수 있다.
-p : 시작포구번호를 지정한다.
-q : 한 집합당 파케트수를 지정한다.
-v : 보다 많은 정보를 보여준다.
-w : 시간경과(timeout)의 값을 지정한다.

하나의 호스트가 망에 련결되여있는가를 검사하는데 사용되는 대표적인 명령이 바로 ping인데 이것은 단순련결성만을 나타낸다. 또한 호스트에서 ICMP파케트를 받지 않게 설정하면 정확한 련결성검사결과를 알수 없다.

traceroute명령은 실지 파케트가 지나가는 경로를 확인하여 호스트에 련결된 망병목(bottleneck)구간을 파악하고 망련결성도 파악할수 있게 하는 명령문이다. traceroute명령에 의해 현재 호스트로부터 kldp.org호스트에 련결된 망경로를 보여주는 실례를 아래에 주었다.

```
kcc@linux:~

# traceroute kldp.org
traceroute to kldp.org (211.169.242.48), 30 hops max, 38 byte packets
 1  61.40.233.1 (61.40.233.1) 0.239 ms 0.185 ms 0.171 ms
 2  10.0.0.1 (10.0.0.1) 1.872 ms 1.921 ms 1.961 ms
 3  211.1707.81 (211.170.7.81) 3.789 ms 8.005 ms 3.866 ms
 4  anybdr3-fe4-1-0.rt.bora.net(210.120.252.169) 3.997ms 4.096ns 3.955ms
 5  any160bb2re0—1-0-1-0.rtbora.net (210.120.193.145) 4.042ms 4.119 ms
 6  210.120.61.34 (210.120.61.34) 4.509 ms 6.596 ms 4.865 ms
 7  210.92.194.214 (210.92.194.214) 4.468 ms 4.449 ms 4.600 ms
 8  211.174.48.86 (211.174.48.86) 4.380 ms 4.439 ms 4.456 ms
 9  211.119.133.234 (211.119.133.234) 5.140 ms 4.596 ms 5.406 ms
10 211.169.242.48 (211.169.242.48) 11.169 ms 11.674 ms 4.830 ms
```

true

/bin/true

참(0)값을 반환

언제나 반드시 참(0)값을 반환하는 명령문이다.

형식 : true[설정값]

—help : 도움말을 보여준다.
—version : 판본을 보여준다.

true명령은 언제나 참값을 반환한다. 쉘안에서 참을 반환할 때에 쓰인다. 즉 참인 경우에 《0》값을 반환한다.

참 고

다중과제처리(Multitasking)

콤퓨터프로그람실행에서의 과제(Task)란 조작체계가 관리하는 프로그람의 기본단위를 말한다.

다중과제처리란 여러개의 과제를 동시에 실행하고 교대로 콤퓨터의 자원을 사용할수 있게 하는것으로서 현재 대부분의 조작체계들에서 지원하고있다. 비협조적다중과제처리(Preemptive Multitasking)에서 매 과제는 상대적중요도, 자원소모량 및 기타 다른 요인들에 따라 그의 우선권순위가 할당된다. 조작체계는 우선권순위가 낮은 값을 가지는 작업의 실행을 정지시킴으로써 우선권순위가 높은 작업을 실행하도록 한다.

협력적다중과제처리(Non-Preemptive Multitasking)는 동시에 여러 과제를 관리하기 위한 조작체계의 기능으로서 반드시 그것들을 먼저 차지하는 능력은 없다.

tty

/usr/bin/tty

말단이름보기

현재 사용중인 말단의 파일이름을 보여준다.

형식 : tty[설정값]

-s,—silent, —quiet : 아무것도 출력하지 않는다.

—help : 도움말을 보여준다

— version : 판본을 보여준다.

tty는 현재 표준입력으로 련결되여있는 말단의 파일이름을 보여준다.

```
kcc@linux:~
# tty
/dev/pts/2
```

참 고

서고(library)

서고는 다른 프로그람들과 련결되기 위하여 만들어진 함수나 보조루틴의 집합으로서 대체로 프로그람을 번역한 다음 함께 련결될수 있도록 번역된 형태의 목적모듈(object module)로 존재한다. 서고는 코드재사용을 위한 목적으로 적용된 초기방법중의 하나로서 많은 다른 프로그람들에서 사용할수 있도록 조작체계나 쏘프트웨어개발환경의 개발자들에 의해 제공되는 경우가 많다. 서고내의 루틴들은 범용적인것도 있으며 3차원동화상처리와 같은 특별한 용도의 서고함수로 제공되는것도 있다.

서고들은 사용자의 프로그람과 련결되여 실행가능한 완전한 프로그람을 이루게 된다. 이러한 련결에는 정적련결과 동적련결이 있다.

tune2fs

/sbin/tune2fs

파일체계설정

ext2파일체계의 첨수를 설정한다.

형식 : tune2fs[설정값]

-c mount회수 : 최대의 마운트(mount)회수를 설정한다.
-e 오유시의 반응 : 적재오유가 발생한 경우 핵심부의 반응을 설정한다.
continue : 정상적으로 계속 실행한다.
remount-ro : 읽기전용으로 마운트한다.
panic : 핵심부공황(kernel panic)을 보여준다.
-g 집단 : 예약된 블로크를 사용할 집단을 설정한다.
-i 파일체계검사간 시간간격[d|m|w] : 매일(d), 매달(m), 매주(w)를 검사시간으로 설정한다. 0인 경우에는 시간검사를 하지 않는다.
-l : 파일체계의 고급블로크(super block)의 내용을 보여준다.
-m 예약블로크퍼센트 : 지정한 장치의 예약블로크퍼센트를 정한다.
-r 예약블로크수 : 지정한 장치의 예약블로크수를 정한다.
-u 사용자 : 예약블로크를 사용할수 있는 사용자를 설정한다.

tune2fs는 파일체계의 고급블로크(super block)정보를 변경한다. 고급블로크는 파일체계의 크기, 사용하는 조작체계 등을 비롯하여 전체적인 파일체계에 대한 정보를 포함한다.

-r설정값을 리용하여 파일체계의 고급블로크정보를 확인할수 있다.

kcc@linux:~

```
# tune2fs -l /dev/hda3
tune2fs 1.23, 15-Aug-2001 for EXT2 FS 0.5b, 95/08/09
Filesystem volume name: <none>
Last mounted on:        <not available>
Filesystem UUID:        d5f36bd2-cf49-42b0-a612-900b4e685ed7
```

```
Filesysiem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      filetype sparse_super
Filesystem state:         not clean
Errors behavior:          Continue
Filesystem OS type :      Linux
Inode count:              1011840
Block  count:             2022181
Reserved block count:     101109
Free blocks:              1418207
Free inodes:              1011331
First block:              0
Block size:               4096
Fragment size:            4096
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         16320
Inode blocks per group:   510
Last mount time:          Fri Jul  5 22:39:34  2002
Last write time:          Fri Jul 19 01:26:34  2002
Mount couut:              4
Maximum  mount count:     25
 Last checked:            Fri Jul 5 21:41:45  2002
 check  interval:         15552000 (6 months)
 Next check after :       Wed Jan 1 21:41:45  2003
 Reserved blocks uid:     0 (user root)
 Reserved blocks gid:     0 (group root)
 Frst inodee:             11
 Inode size:              128
```

U

ul /usr/bin/ul

밑줄긋기

밑줄속성이 있는 문자렬을 밑줄표시(_)와 함께 보여준다.

형식 : ul[option]file

-i : 밑줄속성이 있는 문자의 경우 밑줄문자표시가 아닌 다음행에 밑줄표시로 나타난다. 말단이 밑줄(underline)문자표시를 하지 못하는 경우에 유용하다.

-t 말단 : 출력되는 말단속성을 지정한다.

ul명령문은 밑줄속성의 문자들에 밑줄표시를 하면서 화면에 출력하여 http, ftp등의 주소를 다른 문자렬과 구분되도록 만들어준다. 밑줄의 설정은 /etc/termcap파일을 참조한다.

말단의 종류에 따라 밑줄처리를 못하는 경우, 표준출력으로 보여준다.

【관련명령문】

cat : 표준출력으로 보여준다.

umount

/bin/umount

장치련결의 해제

마운트(mount)한 장치를 해제한다.

형식 : umount[option]장치(파일)
umount[option][-t파일체계속성]장치(파일)

-V : 판본정보를 보여준다.
-h : 도움말을 보여준다.
-y : umount의 내용을 보여준다.
-n : /etc/mtab를 갱신하지 않고 마운트해제한다.
-a : /etc/mtab파일에 모든 파일체계를 마운트해제한다.
-t **파일체계속성** : umount하려고 하는 파일체계의 종류를 지정한다. 두개 이상일 때에는 반점(,)으로 구분한다. no를 접두사로 사용하면 특정 속성의 파일체계를 마운트해제목록에서 제외시킬수 있다.

체계는 파일체계를 사용하기 위해 마운트하며 다시 해제할 때에는 umount를 사용한다.

현재 체계에 마운트된 파일체계는 /etc/mtab에 등록된다.

다음과 같이 마운트되여있는 파일체계를 확인해보자.

《mount -l》을 리용하면 마운트된 목록을 확인하거나 《/etc/mtab》파일을 열어 현재 마운트된 정보를 볼수 있다.

kcc@linux:~

```
# mount -l
/dev/hda1 on / type ext3 (rw) [/]
none on/proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
nane on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda3 on /home type ext2 (rw) []
none on/dev/shm type tmpfs (rw)
/dev/cdrm on /mnt/cdrom type iso9660 (ro,nosuid,nodev,io
```

```
charset=cp949,codepage=949noauto)

# cat /etc/mtab
/dev/hda1 / ext3 rw 0 0
none/proc proc rw 0 0
usbdevfs /proc/bus/usb usbdevfs rw 0 0
none/dev/pts devpts rw,gid=5, mode=620 0 0
/dev/hda3 /home ext2 rw 0 0
none /dev/shm tmpfs rw 0 0
/dev/cdrom /mnt/cdrm iso9660 ro,nosuid,nodev,iocharset=cp949,
codepage=949noauto 0 0
```

우의 가운데서 특정장치의 마운트를 해제할수 있다. 우의 목록에서 보면 /dev/cdrom장치가 /mnt/cdrom에 마운트된것을 확인할수 있다. cdrom 장치를 해제해보자.

```
kcc@linux:~

# umount /mnt/cdrom
# mount -l
# mount -l
/dev/hda1 on /type ext3 (rw) [/]
none on/proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
none on/dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda3 on /home type ext2 (rw) []
none  on /dev/shm type  tmpfs (rw)
```

참 고

마운트(mount)

UNIX 에서는 하드디스크와 영상표시장치, 건반을 비롯한 모든 장치를 파일로 인식한다. 이러한 파일체계들은 뿌리파일등록부(/)의 아래준위파일등록부로 인식되며 나무구조로 구성된다. 이와 같이 장치를 사용하기 위해 뿌리파일등록부의 아래준위 파일등록부에로 붙이는 작업을 마운트(mount)라고 한다.

uname

/bin/uname

체계정보의 보기

체계의 정보를 보여준다.

형식:

-m, —machine : 체계의 하드웨어형식을 보여준다. arch와 같은 기능을 수행한다.
-n, —nodename : 체계의 망마디호스트명을 보여준다.
-r, —release : 조작체계의 출품번호(release number)를 보여준다.
-s. —sysname : 조작체계의 이름을 보여준다.
-v : 조작체계의 판본을 보여준다.
-a, —all : 우의 모든 정보를 보여준다.
—help : 도움말을 보여준다.
—version : 판본을 보여준다.

uname은 체계에 대한 정보를 보여준다.
아래의 실례와 같이 -a설정값을 사용하여 체계의 모든 정보를 볼수 있다.

```
kcc@linux:~
# uname -a
Linux www.linuxroot. co.kp 2.4.13-1hl #1 2001. 11. 04. (일)
04:04:58 KST i686 unknown
Linux : 조작체계이름
www.linuxroot.co.kp : 체계호스트명
2.4.13-1hl : 출품번호
#1 2001. 11. 04. (일) 04:04:58 KST : 조작체계판본
i686 : 체계의 하드웨어형식
```

설정값을 리용하면 개별적정보의 출력도 할수 있다.

참 고

/proc 파일등록부

이 파일등록부는 실지로 디스크에 보관되지는 않는것으로서 주기억기의 정보만을 가지고있다. 즉 실시간적으로 체계의 정보주기억에 담아두는 역할을 한다. 그러므로 이 파일등록부의 파일을 잘 리용하면 체계의 정보를 관리자의 요구에 따라 편리하게 얻을수 있다. 말단에서 ps -aux명령문을 입력하여 확인할수 있는 프로쎄스정보들은 이 파일등록부들을 참고하여 얻어지는것이다.

- cmdline : 프로쎄스를 실행한 명령문을 모두 보여준다.
- cwd : 프로쎄스의 작업파일등록부
- environ : 이 프로쎄스가 참고하는 환경변수정보를 가지고있다.
- exe : 이 프로쎄스를 실행한 2진코드가 기호련결되여있다.
- fd : 프로쎄스가 참조하는 파일목록정보, 파일이름이 0부터 시작되여 련결된 파일을 확인한다.
- maps : 현재 프로쎄스가 사용하고있는 주기억령역정보프로쎄스가 사용하는 파일과 그 파일이 사용하는 주기억령역정보를 볼수 있다.
- mem : 프로쎄스가 호출하는 주기억기
- root : 프로쎄스의 뿌리파일등록부
- stat, statm, status : 프로쎄스의 상태정보들을 담고있다.

/proc/cpuinfo

CPU정보, 체계에서 사용하고있는 CPU의 종류와 성능을 보여준다.

이 파일을 통해 체계의 bogomips수값을 알수 있다

- bogomips : mips(Millions of Instructions Per Second)는 초당 백만회의 명령이라는 뜻으로서 프로그람의 실행속도를 나타내는데 쓰인다. bogo는 bogus(가짜)의 략자이다. 핵심부는 기동시에 핵심부나 매 장치구동기에서 CPU의 속도와 관계없이 일정한 시간을 취할수 있는 지연순환(delay loop)을 필요로 하며 그러한 순환을 위한 보조프로그람을 수행한 결과로 나온 CPU실행속도인 bogomips를 출력하게 된다.

/proc/devices

체계가 사용하는 장치정보를 보여준다.

/proc/dma

DMA(Direct Memory Access)는 체계의 매 장치들이 CPU를 거치지 않고 직접 주기억기에 자료를 보내는 기능으로서 CPU를 거치지 않으므로 체계의 속도를 높일수 있다. /proc/dma는 이 DMA통로의 정보를 보여준다.

/proc/filesytems

핵심부가 지원하는 파일체계의 정보를 보여준다.

/proc/interrupts

매 IRQ(interrupt request:새치기요구)정보를 보여준다.

/proc/ioports

사용하고있는 입출력포구정보를 보여준다.

/proc/kcore

핵심부 주기억 이메지.

/proc/kmsg

핵심부의 동작중에 중요한 사건(Event)이 발생하면 출력하는 통보문을 보여준다.

/prcoc/ksyms

핵심부모듈의 기호정보를 보여준다.

/proc/loadavg

체계의 평균부하를 보여준다.

/proc/meminfo

체계의 주기억정보를 보여준다. 기본주기억, 가상주기억(swap)정보를 보여준다.

/proc/modules

현재 핵심부에 적재된 모듈정보를 보여준다.

/proc/net

망통신규약에 대한 정보를 가지고있다. 이 파일등록부에는 핵심부가 지원하는 각종 통신규약의 현 상태에 대한 정보를 가진 파일들이 있다.

/proc/pci

체계에 있는 PCI장치들의 정보를 보여준다.

/proc/scsi

체계에 있는 SCSI장치들의 정보를 보여준다.

/proc/self

현재 /proc파일등록부를 리용하는 프로세스ID에 기호련결되여 정보를 보여준다.

/proc/stat

핵심부와 체계의 정보들을 보여준다.

/proc/sys

핵심부가 사용중인 핵심부변수정보들을 가지고있다.

/proc/uptime

체계가동시간을 보여준다. uptime명령으로 체계의 가동 시간, 사용인원, 평균체계부하 등의 정보를 볼수 있다.

/proc/version

핵심부판본정보를 보여준다.

uncompress /usr/bin/uncompress

압축해제

compress로 압축된 파일을 푼다.

형식 : uncompress[설정값]압축파일이름

-f : 강제압축을 해제하기전에 같은 이름의 압축파일이 있는 경우 덮쓰기를 한다.
-v : 압축이 해제되는 과정을 보여준다.
-c : 압축해제내용을 표준출력으로 보여준다.
-v : 판본정보를 보여준다.

uncompress는 압축된 파일을 풀어주는 명령문이다.

```
kcc@linux:~

# compress -v mbox
mbox: — replaced with mbox.Z compression: 44.22%
```

-c설정값을 리용하면 압축된 파일을 해제하지는 않지만 해제한 경우의 내용을 미리 보여준다.

```
kcc@linux:~

# uncompress -c mbox.Z
From root Tue Jul 9 03:30:06 2002
Return-Path: <MAILER-DAEMON@pirania.dizikarma.com>
Received: from localhost (localhost)
   by pirania.dizikarma.com (8.11.6/8.11.6) id g68lU6q02258;
   Tue, 9 Jul 2002 03:30:06 +0900
Date: Tue, 9 Jul 2002 03:30:06 +0900
----------------------중간 생략--------------------
 # uncompress -f -v mbox.Z
mbox.Z: —replaced with mbox
```

-f는 현재파일등록부에 같은 이름의 파일이 있는 경우에 덮쓰기를 하도록 하는 설정값이며 -v는 압축해제과정을 보여준다.

unexpand

/usr/bin/unexpand

공백을 tab로 변환

공백을 tab로 변환하는 명령문이다.

형식 : unexpand[설정값]

-a, —all : 모든 공백을 변환한다.

-t 수자, —tabs=수자 : tab크기를 8이 아닌 지정해준 수자로 사용한다.

-t 목록, —tabs=목록 :

-h, —help : 도움말을 보여준다.

-v, —version : 판본을 보여준다.

unexpand는 공백문자를 tab로 변환하여준다.

본문파일에서는 공백건(space bar)으로 입력한 공백문자를 tab로 변환하여 준다. tab는 공백의 8번째에로 유표를 옮기게 되는데 《-t수자》설정값에 의해 tab의 크기를 조절할수 있다.

```
kcc@linux:~

# cat spacefile
tab 10

# unexpand -a -t 10 spacefile
tab10
```

uniq /usr/bin/uniq

중복행의 추출

꼭같은 행이 두번이상 나오는 경우에 하나만 남기고 삭제하는 명령문이다.

형식 : uniq[설정값]

-c, —count : 같은 행이 몇번 나오는가를 표시한다.
-d. —repeated : 중복되여 나오는 행가운데서 한행만을 보여준다.
-D, —all-repeated : 중복되는 모든 행을 보여준다.
-N, —f N, —skip-fields=N : 추출을 무시하려는 행을 지정한다. 시작행부터 N번째 행까지는 검사하지 않는다.
-i, —ignore-case : 중복되는 행들을 한행으로 인식하고 출력한다.
+N, -s N, —skip-chars=N : N번째문자부터 검사를 시작한다.
-u, —unique : 중복행이 없는것만을 보여준다.
-w, —check-chars=N : N번째문자까지 검사한다.
—help : 도움말을 보여준다.

uniq는 중복되는 문자렬을 찾아 추출해내는 명령문이다.

중복되는 문자렬가운데서 중복되는것만을 보여주기, 또는 중복되지 않는것만을 보여주기를 할수 있다. 또한 중복되는 수자를 계수하여 보여줄수도 있다. 검사방법에서는 특정행의 제외, 한행으로부터 지정한 령역까지만을 검사 또는 제외 등을 할수 있다.

다음의 실례를 보자.

last 명령문은 사용자접속에 관한 리력정보를 보여주는 파일이다. uniq를 리용하여 식별이름의 현재 사용상태를 알아보자.

```
kcc@linux:~

# last | sort | uniq -w 1 -c
1
11 dpym      ftpd11239 61.102.14.137   Sat Jul 20 04:02-05:58  (01:55)
11 ftp       tpd2937   61.102.11.143   Wed Jul 10 10:47-10:48 (00:00)
39 hapeanut ftpd1022   211.245.192.136 Sat Jul 6 01:58-01:59  (00,00)
1 knocki     pts/4     168.126.198.46  Sun Jun 23 10:30-10:30 (00:00)
1 newman2    pks/3     61.248.56.57    Mon Jul 22 08:30-08:30 (00:00)
5 orion79    ftpd4066  210.97.3.2      Fri Jun 7 00:25-00:26 (00:00)
75 pirania   ftpd11101 203.245.0.1     Fri Jul 19 21:48-21:48 (00:00)
73 reboot    system boot  2.4.13-1hl   Fri Jul 5 11:37 (00:07)
55 tox2000   ftpd11244 61.73.5.102     Sun Jul 21 01:30-03:49  (02:19)
129 wizard   ftpd11245 211.104.81.226  Sun Jul 21 01:51-04:53 (03:01)
```

last명령으로 보여주는 정보를 sort로 정렬한 다음에 uniq를 리용하여 현사용상태를 계수하였다. -w 1은 첫번째 항목의 단어만 uniq를 사용하여 검사하도록 하며 검사된 결과는 -c설정값에 의해 계수되여 나타난다.

우의 결과로부터 Wizard식별이름이 체계접속(log in) 및 체계탈퇴(log out)의 회수에 있어서 가장 많다는것을 확인할수 있다.

uptime

/usr/bin/uptime

평균부하의 보기

체계의 가동시간과 평균부하를 보여준다.

형식 : uptime[설정값]

-v: 판본정보를 보여준다.

uptime은 평균체계의 부하를 보여준다.

```
kcc@linux:~
# uptime
  7:50pm up 15 days, 21:11, 1 user, load average: 2.00, 2.00, 2.00
```

uptime명령의 입력에 의해 체계가 시작된 시간, 체계가 가동된 시간, 현재 사용자수, 평균부하량(1분, 4분, 15분간, 1시간을 기준으로 평균사용량을 보여준다) 등을 확인할수 있다.

useradd

/usr/bin/useradd

사용자의 추가

체계에 사용자를 추가한다.

형식 : useradd 사용자ID

-c comment : 새로운 사용자의 설명을 추가한다. 사용자의 전체이름을 지정해 줄수 있다.

-d home등록부 : 사용자home식별등록부의 위치를 지정한다. 기본적으로 /home등록부아래에 생성된다.

-e 날자 : 림시사용자의 사용기간을 제한한다. YYYY-MM-DD(년도-월-일)방식으로 지정한다.

-f 남은 날자수 : 림시 사용자의 사용기간은 남은 날자수로 제한된다.

-g 집단 : 새로운 사용자의 집단을 지정한다.

-G 집단 : 새로운 사용자가 포함되는 여러 집단을 지정할수 있다.

-u UserID : 새로운 사용자ID값을 지정한다. 사용자ID값은 /etc/passwd파일에 의해 확인할수 있다.

-p 암호 : 새로운 사용자와 생성과 함께 암호도 지정한다.

-s shel : 새로운 사용자의 쉘을 지정한다.

-m -k skell_dir : skell등록부를 지정한다. 기본적으로 etc/skell등록부의 내용을 새로운 사용자등록부에로 복사한다.

super(특권을 가진)사용자는 useradd명령으로 체계에 새로운 사용자ID를 생성할수 있다. 발급받은 ID로 체계에 접속하여 체계접근(login)을 할수 있다.

```
kcc@linux:~

# useradd newman
# finger newman
Login: newman                  Name: (null)
Directory: /home/newman        Shell: /bin/bash
Never logged in.
No mail.
No Plan.
```

체계관리에서 보면 일시적인 ID를 생성해야 할 때가 있다. 이러한 때에 《-e》설정값으로 날자를 지정해주어 사용한다. 지정한 날자가 지나면 기일이 지난 식별이름의 사용자는 접속할수 없다. 또한 발급한 ID가 늘어날수록 사용자를 구분하기 힘들다. 이러한 때에 -c로 사용자이름을 넣어 구분할수 있다.

```
kcc@linux:~
# useradd -e 2002-07-27 -c ParkJongO newman2
# finger newman2
Login: newman2                              Name: ParkJongo
Directory: /home/newman2                    Shell:/bin/bash
Last login Mon Jul 22 08:30 (KST) on pts/3 from  61.248.56.57
No mail.
No Plan.
```

사용자식별이름을 생성한 후에는 passwd명령을 리용하여 사용자암호를 생성해주어야 한다. -p 설정값을 리용하면 식별이름생성때에 암호도 같이 생성할수 있다. 새로운 식별이름생성시에 /home/사용자식별이름에 /etc/skel 내용이 복사된다. skel등록부를 관리자가 사용자별로 정리하여 복사해올 파일등록부를 지정해줄수도 있다.

```
# useradd -m -k /etc/skel2 -p dkaghek newman3
```

【관련명령문】

userdel : 체계의 사용자를 제거하는 명령문

usermod : 사용자의 설정환경을 변경하는 명령문

userdel

/usr/bin/userdel

사용자의 제거

체계의 사용자 ID를 삭제한다.

형식 : userdel[설정값]

-r : 사용자 ID 뿐만아니라 사용등록부까지 함께 삭제한다.

userdel은 체계의 사용자를 제거하는 명령문이다.

userdel명령만을 사용하는 경우 사용자파일등록부의 정보는 삭제하지 않고 식별이름만이 삭제된다. 사용자파일등록부정보까지 삭제하기 위해서는 -r 설정값을 사용한다.

```
# userdel -r newman
```

일반적으로 사용자식별이름은 《/home/account》로 자기의 home식별이름을 만든다. 물론 사용자생성시에 home등록부위치를 지정할수도 있다.

【관련명령문】

userdel : 체계의 사용자를 제거하는 명령문

usermod : 사용자의 설정환경을 변경하는 명령문

usermod /usr/bin/usermod

사용자설정환경의 변경

현재 사용자의 설정파일을 수정한다.

형식 : usermod[설정값]

-c comment : 사용자의 설명을 변경시킨다. 사용자의 전체이름을 지정해줄수 있다.

-d home등록부 : 사용자 home식별이름등록부의 위치를 지정한다.

-e 날자 : 사용자의 사용기간을 제한한다. YYYY-MM-DD(년-월-일) 방식으로 지정한다.

-f 남은 날자수 : 림시사용자의 사용제한.

-g 집단 : 새로운 사용자의 집단을 지정한다.

-G 집단 : 사용자가 포함되는 여러 집단을 지정한다.

-l 사용자ID : 사용자ID를 변경한다.

-p 암호 : 사용자의 암호를 정한다.

-s shell : 사용자의 쉘을 정한다.

-u userID : 사용자ID값을 변경한다. 사용자ID값은 /etc/passwd파일로 확인할수 있다.

-L : 사용자의 암호를 막아 놓아 접근을 금지한다. 이 설정값은 암호를 지정하는 -p설정값이나 열쇠를 풀어주는 《-U》 설정값과 같이 사용될수 없다.

-U : 사용자접근을 금지시킨 암호를 풀어준다. 이 설정값은 암호를 지정하는 p설정값이나 암호잠금(password lock)을 걸어주는 《-L》 설정값과 같이 사용될수 없다.

usermod는 사용자의 정보를 수정할수 있다. 이미 있던 사용자정보를 변경하여 체계에서 식별이름에 대한 관리를 편리하게 할수도 있다.

주의할 점은 다음과 같이 체계접속중의 사용자정보는 수정하지 못한다는것이다. newman ID에 대한 사용자이름과 암호를 변경해보자.

```
kcc@linux:~

# usermod -c PakCholHo-p whdrud  newman
# finger newman
Login: newman                Name: PakCholHo

Directory: /home/newman2  Shell:/bin/bash
Last login Mon Jul 22 08:30 (KST) on pts/3 from
61.248.56.57
No  mail.
No Plan.
```

사용자접속 ID를 변경할수 있다. ID만 변경하고 다른 설정은 변경하지 않는다. 만약 home등록부도 변경하려고 한다면 파일등록부를 생성하고 《-d》설정값으로 따로 변경해주어야 한다. newman이라는 ID를 oldman으로 변경하여 보자.

```
kcc@linux:~

# usermod -l oldman newman
# finger oldman
Login: oldman                 Name: PakCholHo
Directory: /home/newman     Shell: /bin/bash
Laet login Mon Jul 22 08:30 (KST) on pts/3
from 61.248.56.57
No mail.
No Plan.
```

접속을 허용하고있는 사용자 ID중에서 사용자정보와 파일들은 삭제하지 않고 접근만을 금지시키려고 하는 경우에 잠금을 걸어준다.

```
# usermod -L newman
```

다시 접근을 허용하려는 경우에는 《-U》설정값으로 잠금을 풀어준다.

```
# usermod -U newman
```

usermod명령은 -e설정값의 사용에 의해 림시사용자의 사용기한을 정해줄 수 있다. 2004년 7월 28일까지 newman이라는 ID를 가지는 사용자가 접속할수 있도록 설정해보자.

```
# usermod -e 2004-07-28 newman
```

users

/usr/bin/users

사용자보기

현재 체계를 사용하는 사용자를 보여준다.

형식 : users [설정값]

—help : 도움말을 보여준다.
—version : 판본을 보여준다.

체계에 접속하고있는 사용자목록을 보여준다.

```
kcc@linux:~
# users
pirania pirania root root
```

체계에 접속한 수자만큼 보여준다. 우의 화면출력에서 보면 이 체계에는 현재 pirania식별이름으로 2개, 뿌리식별이름으로 2개가 접속되여있다는것을 알수 있다. 보다 상세한 정보를 보려는 경우에는 finger, w, who 등의 명령문을 사용한다.

【관련명령문】

finger : 체계에 접속한 사용자정보를 보여준다.
w : 체계에 접속한 사용자정보를 보여준다.
who : 체계에 접속한 사용자정보를 보여준다.

W

w /usr/bin/w

현재사용자의 보기

현재 체계접속한 사용자의 정보를 보여주는 명령문이다.

형식 : w [설정값][사용자]

-f : 원격접속한 호스트의 이름을 보여주지 않는다.

-h : 매 항목에 대한 항목명(field name)을 생략한다.

-s : 간략화한 형식으로 정보를 보여준다.

-V : 판본정보를 보여준다.

현재 체계접속한 사용자정보를 보관해둔 /var/run/utmp파일의 정보를 보여준다. who명령과는 약간 달리 WHAT항목이 있어 사용자가 현재 접속하여 진행하고있는 작업내용을 볼수 있다.

```
kcc@linux:~
# w
2:01am up 16:02, 4 users, load average: 0.00, 0.00, 0.00
USER TTY     FROM            LOGIN@  IDLE   JCPU  PCPU  WHAT
root  pts/10  -              6:38pm  7:19m  0.20s 0.02s nslookup
root  pts/1   211.49.155.165 9:34pm  0.00s  0.56s 0.05s w
root  pts/2   61.40.233.218  1:56am  4:35   0.05s 0.05s -bash
```

wall

/usr/bin/wall

모든 사용자에게 통보문 보내기

모든 말단에 통보문을 출력하는 명령문이다.

형식 : wall[통보문]

체계를 사용하는 모든 말단에 통보문을 보내는 명령이다. 이 경우 현재 체계접속한 사용자들은 Wall명령으로 보내온 통보문을 보고 작업하던 내용을 완료하게 된다.

wall명령뒤에 원하는 통보문을 입력하면 매 말단에서 작업중인 사용자들은 Broadcast message로 시작하는 통보문과 함께 입력한 통보문을 볼수 있다.

```
kcc@linux:~

# wall After 5 minites, System is reboot!!!!
#
Broadcast message from root(pts/1) Fri Jun 21 02:11:45 2004...

After 5 minites, System is reboot!!!!
```

[주의]

shudown에서 완료할 때 이 명령을 사용한다.

wc /usr/bin/wc

단어수의 계산

문서가 몇행, 몇단어로 이루어져있는가를 한번에 계산해준다.

형식 : wc [설정값][파일]

-c, -bytes, —chars : 문자수만을 보여준다.
-l, —lines : 행수만을 보여준다.
-w, —words : 단어수만을 보여준다.
—help :도움말을 보여준다.
—version : 판본정보를 보여준다.

wc명령은 파일의 행, 단어, 문자의 수를 한행에 화면출력할수 있게 한다. 아래의 실례에서는 httpd.conf파일이 1467개의 행과 7304개의 단어, 51633개의 문자로 이루어져 있다는것을 보여준다.

```
kcc@linux:~
# wc /etc/httpd/conf/httpd.conf
1467  7304  51633 /etc/httpd/conf/httpd.conf
```

-L설정값은 가장 긴행의 길이만을 나타낸다.

```
kcc@linux:~
# wc -L /etc/httpd/conf/httpd.conf
105 /etc/httpd/conf/httpd.conf
```

아래의 실례에서는 apache사용자소유의 프로쎄스개수를 보여준다. ps -u 설정값은 지정한 소유자의 프로쎄스만을 보여주는 명령이다. 이것을 파이프(|)로 받아 wc -l명령이 행수를 출력하게 된다.

결과 수값 18은 현재 apache소유자의 프로쎄스개수가 18개라는것을 나타낸다.

```
kcc@linux:~
# ps -u apache | wc-l
18
```

현재 체계접속한 사용자수를 장악해보자. who명령은 현재 체계접속한 사용자정보를 보여주는것으로서 아래의 례에서는 wc -l명령과 함께 사용하여 현재 체계접속한 사용자수를 보여준다.

아래에서 보면 이 호스트에는 현재 4명의 사용자가 접속하고있다.

```
kcc@linux:~
# who | wc-l
4
```

whatis

/usr/bin/whatis

명령검색

whatis DB를 검색하여 실마리어(Keyword))에 해당하는 명령문만 보여준다.

형식 : whatis 실마리어

apropos는 whatis DB를 검색하여 관련명령문과 그 명령문에 대한 간단한 설명을 보여준다. whatis는 whatis DB에서 명령문만 검색하여 출력하지만 apropos는 명령문과 설명의 모두를 검색하여 결과를 보여준다.

이것은 man-k명령과 같다.

```
kcc@linux:~

# apropos who
at.allow [at]   (5) - determine who can submit jobs via at or batch
at.deny [at]    (5) - determine who can submit jobs via at or  batch
ftpwho          (1) - show current process information for each  ftp user
fwhois[whois]   (1) - query a whois or nicname database
rusers          (1) - who is logged in to machines on local  network
rwho            (1) - who is logged in on local machines
rwhod           (8) - system status server
w               (1) - Shaw who is logged on and what they are doing
who             (1) - show who is logged on
whoami          (1) - print effective userid
whois           (1) - query a whois or nicname database
whom            (1) - report to whom a message would go
# whatis who
who             (1) - show who is logged on
```

【관련명령문】

apropos whatis : 자료기지(DB)를 검색하여 관련명령문을 보여줌

whereis : 실행파일, 원천파일, 안내문서의 위치를 검색

whereis

/usr/bin/whereis

명령문경로의 검색

명령문을 입력받아 해당 명령문의 절대경로를 찾아준다.

형식 : whereis [설정값]파일이름

-b : 2진파일만을 찾는다.
-m : 안내문서(manual)페지만을 찾는다.
-s : 원천파일만을 찾는다.
-u : 일반적인것이 아닌 항목을 찾는다. 이 설정값은 보통 특정파일을 제외하는데 사용된다.
-B : 실행파일을 찾을 경로를 지정한다.
-M : 안내문서페지를 찾을 경로를 지정한다.
-S : 원천파일을 찾을 경로를 지정한다.
-f : 경로지정의 다음에 오는것은 찾으려는 파일이름을 알리는 설정값이다. 이 설정값은 -B, -M, -S 등의 설정값들이 사용될 때에 함께 사용해야 한다.

whereis는 실행파일의 위치와 함께 원천파일, 설정파일, 안내문서페지를 검색하여 보여준다. 이것은 locate(slocate)명령에 비해 검색속도가 느리고 검색범위의 제한을 받으며 출력결과를 단순하게 보여준다.

which명령은 $PATH내의 실행파일의 위치를 알려준다. 이와 반면에 whereis는 $PATH내의 실행파일경로뿐만 아니라 안내문서페지의 전체 경로도 알려준다.

아래와 같이 whereis명령으로 rm명령의 실행파일위치와 안내문서페지(man페지)를 한번 검색해보자.

```
kcc@linux:~
# whereis rm
rm: /bin/rm /usr/share/man/man1/ rm.1.gz
```

```
kcc@linux:~

# which rm
alias rm='rm -i'
/bin/rm
```

【관련명령문】

locate(slocate) : 자체자료기지를 통해 전체 파일등록부에서 파일을 찾아주는 명령문

which : $PATH환경변수내의 파일위치를 찾아주는 명령문

which

/usr/bin/which

파일경로의 검색

$PATH환경변수내의 파일위치를 알려주는 명령문이다.

설정값 : which [설정값][--][명령문][...]

-a, —all : 모든것을 보여준다.
-i, —read-alias : alias설정환경을 보여준다.
—skip-alias : alias설정을 무시한다.
skip-dot : dot(.)로 시작하는 등록부를 제외한다.
—skip-tilde : 물결표문자(~)로 시작하는 등록부($HOME등록부)를 제외한다.
—show-dot : dot(.)으로 시작하는 등록부를 포함한다.
—show-tilde : 물결표문자(~)로 시작하는 등록부를 포함한다.
-v, -V, —version : 판본정보를 보여준다.

PATH환경변수에 있는 파일등록부를 모두 검색하여 찾으려고 하는 명령문의 절대경로(full path)를 보여준다. 아래의 실례에서 보면 which는 alias설정이 되여있으며 /usr/bin/which에 위치한다.

```
kcc@linux:~
# which which
alias which='alias | /usr/bin/which —tty-only —read-alias —
show-dot —show-tilde'
/usr/bin/which
```

who

/usr/bin/who

사용자보기

현재 호스트에 접속한 사용자정보를 보여주는 명령문이다.

형식 : who [설정값]..[파일 | 인수1 인수2]

-m : who am i와 같다.
-q, —count : 접속명과 접속회수를 출력한다. 다른 설정값은 무시된다.
-s : 무시된다. 다만 다른 판본의 who와 호환되도록 하기 위해서이다.
-i, -u, —idle : 체계접속 후, 사용자가 얼마의 시간, 분동안 한가하게 있었는가를 보여준다. 《.'》은 바로 방금전까지도 활동을 하고있다는 의미이다. 《old》는 24시간이상 한가하게 있었음을 의미한다.
-H, —heading : column header line을 출력한다.
-w, -T, —mesg, -message, —writable : -s 와 마찬가지로 체계접속명(login name)다음에 사용자의 통보문상황을 문자로 출력한다.

+ : write통보문을 허가
− : write통보문을 허가하지 않음
? : 말단장치를 찾을수 없음
—help : 사용법을 출력한다.
—version : 판본정보를 출력한다.

Who는 현재 접속한 사용자를 보여주는것으로서 utmp파일의 보관된 정보를 출력한다. 사용자가 원격호스트에 접속할 때에 utmp에 체계접속한 사용자정보를 기록하며 사용자가 원격호스트로부터 접속탈퇴할 때에는 기록되여있는 정보를 삭제하게 된다.

RedHat배포판에서 utmp파일은 /var/run/utmp에 존재한다.

who명령은 현재 호스트에 접속해있는 사용자의 체계접속명, 말단, 접속시간, 원격호스트 또는 X display를 보여준다.

```
kcc@linux:~

# who
root     tty1     Apr 26 10:02
root     :0       May 3 14:17
root     pts/2    May 7 12:01
root     pts/1    May 7 22:27 (211.49.155.249)
root     pts/0    May 7 20:50 (211.49.155.249)
```

who /var/log/wtmp는 last명령과 같다.

last는 /var/log/wtmp파일의 정보를 보여주는 명령으로서 체계가 기동된 때로부터 현재까지의 체계접속, 접속탈퇴에 관한 사용자정보가 기록되여 있다.

```
kcc@linux:~

# who am i
ns.linuxroot.co.kp!root     pts/0    May  7 20:50
(211.49.155.249)
```

who명령에 2개의 am i 인수를 첨부해주면 현재 체계접속한 호스트이름 또는 IP주소의 다음에 !를 출력하며 이어서 체계접속명, 접속한 말단, 접속시간 등을 보여준다.

【관련명령문】

last : 체계가 기동된 때로부터 현재까지의 체계접속, 접속탈퇴에 관한 사용자정보를 보여주는데 /var/log/wtmp파일을 참고한다.

lastlog : /etc/passwd파일에 등록된 사용자가 가장 마지막에 체계접속한 시간정보를 보여준다.

users : 현재 호스트에 접속한 사용자이름을 출력한다.

w : who와 비슷한 역할을 하는것으로서 현재 체계접속한 사용자 및 사용자의 현재작업을 보여준다.

whoami : 현재 접속하고있는 쉘의 사용자 ID를 확인한다.

whoami

/usr/bin/whoami

유효사용자의 보기

현재 체계접속한 사용자 ID를 보여준다.

형식 : whoami

사용자명에 련관된 현재의 사용자 ID를 보여준다. 이것은 id -un명령과 같다. 이 명령은 처음 체계접속한 다음 su명령에 의해 다른 사용자로 접속하여 사용하다가 현재 사용하는 식별이름이 어느 식별이름인가를 확인할 때에 사용한다. id명령은 현재 사용자의 실제 ID와 유효사용자 ID, 집단 ID까지 모두 보여주지만 whoami는 체계접속한 사용자의 ID만을 보여준다.

```
kcc@linux:~

$ whoami
admin

$ id
uid=500(admin) gid=500(admin) group=500(admin)

$ id -un
admin
```

write

/usr/bin/write

다른 사용자에게 통보문 보내기

명령행에서 간단한 통보문을 보내는 명령문이다.

형식 : write 사용자 [tty]

write는 간단하게 호스트에 접속해있는 사용자에게 통보문을 보낼수 있게 한다. 이것은 여러 통신이나 인터네트 싸이트에서 제공하는 알림보내기 기능과 같은것이라고 할수 있다.

admin사용자가 뿌리관리자에게 통보문을 보내는 경우를 실례로 들어보자. 통보문을 끝낼 때는 [Ctrl+D] 를 입력하면 된다.

```
kcc@linux:~
$ write root
This is test
```

아래의것은 linuxtoot.co.kp의 호스트root에서 나타나는 통보문이다. 통보문의 끝은 EOF이다.

```
kcc@linux:~
#
Message from admin@ns.linuxroot.co.kp on pts/1 at 03:17 ...
This is test
EOF
```

【관련명령문】

talk : 대화형통보문 보내기

X

xargs /usr/bin/xargs

표준입력을 명령으로 변환

첨수들을 추출하여 넘겨주고 그 명령문을 다시 조합하여 처리하는 명령문이다.

형식 : xargs[설정값][명령문]

-eof[=eof-str], -e[eof-str] : EOF 또는 지정된 문자렬이 나올 때까지 계속 xargs를 통한 추출을 진행한다.

-help : 도움말을 보여준다.

-replace[= replace-str], -i[replace-str] : 각 ReplaceString 을 Argument에 삽입하면서 표준입력의 각 행을 Command매개변수의 인수로 삽입한다.

-max-lines[=max-lines], -l[max-lines] : 비공백매개변수행의 지정된 수와 함께 Command매개변수는 표준입력으로부터 읽는다.

-max-args=max-args, -n max-args : Number매개변수에 의해 최대로 지정될 때까지 가능한한 많은 표준입력인수를 사용하여 Command매개변수를 수행한다.

-interactive, -p : Command매개변수의 수행여부를 묻는다. 또한 -p가 -t를 동작시킨다.

-max-chars=max-chars, -s max-chars : 지령행마다의 최대문자수를 제한한다.

-verbose, -t : 실행과정을 자세히 보여준다.

-version : 판본을 보여준다.

-exit, -x : 지령행이 -s에 의해 지정한 바이트보다 큰 경우에 xargs를 완료한다.

xargs는 명령문에 첨수들을 추출하여 넘겨주며 명령문자체를 재조합한다. 자기가 받아들인 자료를 추출용으로 분리하여 명령문이 처리할수 있게 한다.

xargs명령을 사용한 실례들을 아래에 주었다.

ㄱ) 현재 파일등록부의 모든 파일을 하나씩 확인하면서 .gz파일로 압축해보자.

```
# ls l xargs -p -l gzip
```

ㄴ) 현재 파일등록부의 파일을 한행에 10개씩 나타내 보자.

```
# ls l xargs -n 10 echo
```

ㄷ) Apache와 관련된 패키쥐를 하나의 명령조합으로 삭제해보자.

```
# rpm -qa lgrep apache  |  xargs rpm   e --nodeps
```

Y

ypbind /usr/bin/ypbind

NIS구동

NIS결합프로세스로 NIS봉사기와 의뢰기를 련결한다.

형식 : ypbind[설정값]

-broadcast : 지정한 NIS봉사기에 결합(binding)을 위해 정보를 방송(broadcasting)한다.

-ypset : 원격호스트의 ypset지령을 허가한다. 이것은 보안상 매우 위험하므로 오유수정할 때에만 사용한다.

-ypsetme : ypset요청을 국부호스트에만 허가하는가를 나타낸다.
이것은 부정한 파케트에 의해 망체계를 파괴할수 있으므로 사용제한해야한다.

-c : 설정파일에 오유가 있는가를 검사한다.

-debug : 오유수정방식

-f configfile : 기본설정파일인 /etc/yp.conf파일을 사용하지 않고 지정한 설정파일을 사용한다.

-p port : 지정한 포구를 사용한다.

—version : 판본정보를 보여준다.

ypbind는 NIS(Network Information Service/System)봉사기결합봉사를 위한 /etc/yp.conf설정파일을 가지고있다.

의뢰기가 NIS map로부터 정보를 요청하면 ypbind데몬은 이것을 망을 통해 방송(broadcasting)한다. 봉사기가 응답하면 의뢰기는 데몬에게 인터네트주소와 포구번호정보를 넘겨준다.

참 고

결합(bind)

RPC를 사용할 때에 결합(bind)한다는것은 의뢰기프로그람이 작업요청을 할수 있도록 원격지봉사기프로그람을 위치시키는것을 말하며 이것은 보통 중앙에 유지되고있는 접근가능한 망봉사기프로그람들의 파일등록부를 호출함으로써 이루어진다.

MySQL의 기본사용법

① mysql뿌리사용자에 대한 암호설정

```
mysql>update user set password=password('rootpasswd')where useruser='root';
```

이 명령을 수행한 후에는 MySQL봉사기를 다시 시작해야 한다.

② 새로운 자료기지를 생성

```
# mysqladmin -u root -p create sample
```

③ 자료기지의 삭제

```
# mysqladmin -u root -p drop sample
```

④ 생성된 자료기지의 등록

```
mysql>insert into db values('%','sample','php', ,'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y',);
```

⑤ 새로운 사용자(자료기지소유자)의 등록

```
mysql>insert into user (host,user,password) values('localhost
','php',password('phppasswd'));
```

⑥ sample자료기지의 예비보관

```
# mysqladmin -u php -p sample > sample_backup.sql
```

⑦ 예비보관된 sample자료기지의 복구

```
# mysql -u php -p sample < sample_backup.sql
```

ypcat

/usr/bin/ypcat

NIS련결의 확인

NIS자료기지의 모든 코드(key)값을 보여주는 명령문이다.

형식 : ypcat[설정값]mapname

-d domain : domainname지령에 의해 나타나는 령역을 사용하지 않고 지정된 령역을 사용한다.
-k : map key를 보여준다.
-k : mapname을 map이름으로 번역하지 않는다.
-x : map별명번역표를 보여준다.

NIS봉사기를 설정한 경우에 이것을 검사확인할수 있어야 한다. ypcat명령은 기본설정파일을 만들고 이것이 봉사기에 련결할수 있는가를 검사확인한다.

ypcat -x명령으로 map별명번역표를 한번 보자.

```
kcc@linux:~
# ypcat -x
Use "ethers"    for  map  "ethers.byname"
Use "aliases"   for  map  "mail.aliases"
Use "services"  for  map  "services.byname"
Use "protocols" for  map  "protocols.bynumber"
Use "hosts"     for  map  "hosts.byname"
Use "networks"  for  map  "networks.byaddr"
Use "group"     for  map  "group.byname"
Use "passwd"    for  map  "passwd.byname"
```

hosts.byname을 지정하여 봉사기에 제대로 련결되는가를 확인해볼수 있다.

```
# ypcat hosts.byname
```

망전체의 암호 map인 passwd.byname을 보려면 map이름인 passwd를 지정한다.

```
# ypcat passwd
```

참 고

RPC(Remote Procedure Call: 원격절차호출)

RPC는 한 프로그람이 망우의 다른 콤퓨터에 위치하고있는 프로그람에 봉사를 요청하는데 사용되는 통신규약으로서 이때에 봉사를 요청하는 프로그람은 망에 대한 상세내용을 알 필요가 없다(절차호출이란 때로는 함수 또는 루틴호출의 의미로도 쓰인다). RPC는 의뢰기 및 봉사기방식을 사용하는데 봉사를 요청하는 프로그람이 의뢰기이며 봉사를 제공하는 프로그람이 봉사기이다. 다른 정상적인 또는 자체적인 절차의 호출과 마찬가지로 RPC도 요청하는 프로그람이 원격절차의 처리결과가 반환될 때까지 일시 정지되여야 하는 동기운영방식을 취한다. 그러나 가벼운 프로세스의 사용이나 같은 주소공간을 공유하는 쓰레드 등은 여러개의 RPC들을 동시에 수행될 수 있도록 허용한다.

ypchfn /usr/bin/ypchfn

사용자정보의 변경

NIS봉사기의 사용자정보를 변경하는 명령문이다.

형식 : ypchfn[사용자]

ypchfn는 RPC를 통해 호스트의 yppasswdd데몬에 접속하여 사용자이름, 주소, 전화번호 등의 정보를 변경하는 명령문으로서 finger명령으로 보여지는 정보이다. 이것은 yppasswd -f명령과 같다.

아래의것은 ypchfn명령을 실행하였을 때에 보여주는 항목들이다. 항목을 입력하지 않으려면 Enter 또는 사용자정보를 입력한다.

아래에서는 ypchfn명령만 입력하여 체계접속한 사용자의 정보를 변경하였다.

```
kcc@linux:~

# ypchfn
Name[]:
Location []:
Office Phone []:
Home Phone []:
```

【관련명령문】

ypchsh : NIS봉사기의 사용자쉘을 변경한다.

yppasswd : NIS봉사기의 사용자암호 및 사용자정보들을 변경한다.

ypchsh

/usr/bin/ypchsh

쉘의 변경

NIS봉사기의 사용자쉘을 변경하는 명령문이다.

형식 : ypchsh [사용자]

ypchsh명령은 보통 chsh명령과 같은 기능을 하는것으로서 RPC를 통하여 호스트의 yppasswdd암호데몬에 접속하여 쉘을 변경할수 있다.

이것은 yppasswd -1명령과 같다.

아래의것은 ypchsh명령을 실행하였을 때에 보여주는 내용이다. 현재 기본설정된 쉘은 /bin/bash로 나타나고있으며 변경하려는 쉘을 입력하면 된다.

```
kcc@linux:~
# ypchsh
Password:
New shell [/bin/bash]:
```

【관련명령문】

ypchfn : NIS봉사기의 사용자정보를 변경한다.
yppasswd : NIS봉사기의 사용자암호를 변경한다.

ypmatch

/usr/bin/ypmatch

key값의 보기

MIS map내에서 주어진 key값을 보여주는 명령문이다.

형식 : ypmatch[설정값] key...mapname
ypmatch -x

-d domain : domainname지령으로 보여주는 기본도메인을 사용하지 않고 지정한 domain을 사용한다.
-k : map key를 보여준다.
-t : map이름을 별명으로 변역하지 않는다.
-x : map별명표를 보여준다.

ypmatch명령은 N1S map에서 하나이상의 key와 련관된 값을 보여준다. 아래의것은 ypmatch -x명령으로 보여주는 map별명표이다.

```
kcc@linux:~

# ypmatch -x
Use "ethers"    for map "ethers.byname"
Use "aliases"   for map "mail.aliases"
Use "services   for map "services.byname"
Use "protocols" for map "protocols.bynumber"
Use "hosts"     for map "hosts. byname"
Use "networks"  for map "networks.byaddr"
Use "group"     for map "group.byname"
Use "passwd"    for map "passwd.byname"
```

그러면 그중에서 nis령역에 있는 hosts map의 host1 key를 보도록 하자.

```
# ypmatch -d nis -k  host1 hosts
```

-d설정값은 지정한 령역에 대한 정보를 -k설정값으로 지정한 map key를 보여주게 되는것이다.

참 고

su사용자를 제한하기

보안을 위해 지정한 사용자가 아니면 su를 사용하지 못하는것과 동시에 리력을 남기도록 설정해보자. 먼저 etc/pam.d/su의 첫행에 다음과 같이 추가한다.

```
auth required /lib/security/pam_wheel.so debug group=wheel
```

/etc/group의 wheel집단에 su -허용하려는 사용자집단을 등록한다.

```
wheel:x:10:root,someone
```

만약 허용되지 않은 집단의 사용자가 su -를 시도하면 /var/log/message에 PAM-wheel[25873]: Access denied for ‘babo’ ‘to’ ‘root’와 같이 리력이 기록된다.

yppasswd

/usr/bin/yppasswd

암호의 변경

NIS봉사기의 사용자암호를 변경하는 명령문이다.

형식 : yppasswd[설정값][사용자]

-f : ypchfn지령과 같다.
-l : ypchsh지령과 같다.
-p : 암호를 변경한다.
-?, —help : 도움말을 보여준다.
—usage : 간단한 사용법을 보여준다.
—version : 판본정보를 보여준다.

보통 체계에서는 NIS암호와 집단파일을 일반적으로 /var/yp/ypfiles/에 보관한다. 이것을 사용하는 경우에는 일반적인 암호정보나 변경을 하는 명령인 passwd, chfn, adduser등을 사용할수 없다.

이러한 경우에 yppasswdd의 암호데몬을 기동시킨 다음에 yppasswd, ypchsh, ypchfn명령으로 이와 같은 사용자암호정보를 변경할수 있다.

admin사용자의 NIS암호를 변경하려면 다음과 같이 한다.

```
kcc@linux:~
# yppasswd admin
New password:
Retype new password:
passwd: all authentication tokens updated successfully
```

yppasswd -f명령은 ypchfn명령과 같은 기능을 수행한다.

```
# yppasswd -f admin
```

yppasswd -l명령은 ypchsh명령과 같은 기능을 수행한다.

```
# yppasswd -l admin
```

yppoll /usr/bin/yppoll

NIS정보의 보기

master봉사기의 NIS map와 판본을 보여주는 명령문이다.

형식 : yppoll [설정값] mapname

설정값

-d domain : domainname명령으로 보여주는 기본령역을 사용하지 않고 지정한 령역을 사용한다.

-h host : 기본봉사기가 아닌 지정한 host 를 사용한다. 기본봉사기는 ypwhich 명령으로 찾는다.

yppoll명령은 ypserv데몬을 사용하여 지정한 mapname의 NIS map와 판본에 대한 정보를 보여준다.

yppush

/usr/bin/yppush

NIS전체의 반영

NIS봉사기의 변경된 자료기지가 적용되도록 하는 명령문이다.

형식 : yppush [설정값] mapname

-d domain : domainname지령으로 보여주는 기본도메인을 사용하지 않고 지정한 domain을 사용한다.

-h host : 기본봉사기가 아닌 지정한 host 를 사용한다. 기본봉사기는 ypwhich 지령으로 찾는다.

yppush명령은 master봉사기의 변경된 자료기지가 slave봉사기에 적용되도록 하는 명령문이다. slave봉사기들은 yppush명령을 통해 변경사항을 검사확인하며 변경된 자료기지로부터 필요한 변경사항들을 갱신할수 있다.

ypserv

/usr/bin/ypserv

NIS봉사기

NIS봉사기프로그람이다.

형식 : ypserv [설정값]

ypserv는 NIS(Network Information Services)봉사기에서 의뢰기로 방송하면서 봉사기의 위치를 알리고 의뢰기들의 요청을 받아 응답처리하는 역할을 한다.

이에 대한 설정파일은 /etc/ypserv.conf파일이다. 기본적으로 설정된 내용을 특별히 수정할것은 없다.

```
kcc@linux:~

# cat /etc/ypserv.conf
#
# Some options for ypserv. This things are all not needed, if
# you have a Linux net.
dns: no

# Not everybody should see the shadow passwords, not  secure, since
# under MSDOG everbody is root and can access ports < 1024 !!!!
*                         : shadaw.byname   : port   : yes
                          : passwd.adjunct.byname   : port   : yes

# a rule for them above, that's much faster.
:*                           :*                : none
```

/var/yp/securenets파일이 있는 경우에 ypserv명령은 파일에 지정된 IP범위내에서만 호스트에 응답한다. 적절한 망범위를 지정하여 수정해준다.

```
kcc@linux:~
# cat securenets
#
# Always allow access for localhost
255.0.0.0                   127.0.0.0

# This line gives access to everybody. PLEASE ADJUST!
0.0.0.0         0.0.0.0
```

모든 설정이 되였으면 ypserv데몬을 실행한다.

```
# /etc/rc.d/init.d/ypserv start
```

Z

zcat /usr/bin/zcat

압축된 파일의 내용을 보기

압축되어있는 본문파일의 내용을 볼수 있도록 하는 명령문이다.

형식 : zcat[설정값][파일...]

-c. —stdout : 압축파일을 변환하지 않고 기본출력으로 보여준다.
-d, —decompress : 압축을 해제한다.
-f,—force : 출력파일을 강제로 덧쓰기한다.
-h, —help: 도움말을 보여준다.
-l, —list : 압축된 파일들의 목록을 보여준다.
-L, —license : 쏘프트웨어사용허가권을 보여준다.
-n, —no-name : 원본파일의 이름, 시간을 보관하지 않는다.
-N, —name : 원본파일의 이름, 시간을 보관한다.
-q, —quiet : 모든 경고를 보여주지 않는다.
-r, —recusive : 하위등록부까지 회귀적으로 동작한다.
-S .suf -suffix .suf : .suf확장자로 압축한다.
-t —test : 압축된 파일을 그대로 검사한다.
-v —verbose : 상세하게 보여준다.
-1 —fast : 압축을 보다 빠른 속도로 한다.
-9 — best : 압축률을 높인다.

zcat명령은 gzip, compress 등의 압축파일의 내용을 볼수 있다. 이것은 gunzip -c명령과 같다.

아래의것은 $HOME/.bash_profile을 test.tar.gz파일로 압축한 다음 zcat으로 파일내용을 출력하도록 한것이다. 정상적으로 gz파일의 내용을 볼수 있다.

```
kcc@linux:~

# zcat test.tar.gz
test01006440000000000000000000000352074672337300010472 0ustar
rootroot# .bash_profile

# Get the aliases and functions
if [-f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
BASH_ENV=$HOME/.bashrc
USERNAME="root"

export USERNAME BASH_ENV PATH
```

-l설정값은 압축된 파일의 목록과 정보를 볼수 있게 한다.

```
kcc@linux:~

# zcat -l test.tar.gz
campressed   uncompressed  ratio uncompressed_name
         288     10240   97.3% test.tar
```

zcmp, zdiff

/usr/bin/zcmp

압축된 파일의 내용비교

압축된 파일을 비교하는 명령문이다.

형식 : zcmp [설정값]파일1 파일2

zcmp, zdiff는 압축되어있는 파일들간에 파일비교를 하는 명령이다.

이 명령을 리용하면 지시된 설정값이 모두 cnf 혹은 diff명령으로 직접 전달되여 압축을 해제하지 않고도 파일을 비교하거나 파일간의 차이점을 확인할수 있다.

만일 하나의 파일만 지정하면 지정한 file1.gz의 자동으로 압축을 풀어 file1파일과 비교한다. 매 2개파일을 지정하면 압축된 파일을 자동적으로 압축해제하여 cmp 또는 diff명령에 직접 전달하게 된다.

```
kcc@linux:~
# zcmp file1.gz file2.gz
- /tmp/gz.sByTng differ: char 17, line 2
# zdiff file1.gz file2.gz
2,7d1
<
< # Get the aliases and functions
< if [ -f ~/.bashrc ]; then
<         . ~/.bashrc
< fi
<
```

파일을 비교한 결과 file 1.gz의 내용중 2번째로부터 7번째행까지의 내용은 file2.gz파일에서 찾을수 없다는것을 알수 있다.

【관련명령문】

cmp : 파일비교명령문(diff명령을 간단화한것)

diff : 두 파일간의 차이점에 대해 보여주는 명령문

zdump

/usr/bin/zdump

매 시간대에 의한 시간보기

명령행의 알려진 시간대(time zone)의 목록에서 시간을 보여주는 명령문이다.

형식 : zdump [설정값][zones]

-v : 자세한 내용을 보여준다.

주어진 시간대에서 해당한 시간을 화면에 출력한다.

```
kcc@linux:~
# zdump EST
EST Sun Jun 23 09:00:44 2004 EST
```

참 고

파일체계를 모두 복사하기

어떤 파일체계의 전체 혹은 나무구조의 파일등록부를 /copy라는 파일등록부에로 동일하게 복사하기 위해서는 다음과 같이 실행한다.

```
kcc@linux:~
# mkdir /copy
# cd /targetdir
# find . -depth -print | cpio -pmdvl /copy
```

우와 같이 하면 /targetdir의 모든것이 /copy에로 복사된다.

zmore

/usr/bin/zmore

압축된 본문파일의 내용보기

압축파일의 내용을 more명령으로 보여주는 명령문이다.

형식 : zmore[파일...]

zmore명령은 압축된 파일의 내용을 more명령의 형식으로 볼수 있게 한다.

아래의것은 $HOME/.bash_profile을 test.tar.gz형식으로 압축한 다음 이것을 zmore명령으로 화면출력한 결과이다.

```
kcc@linux:~

# zmore test.tar.gz
test0100644000000000000000000000003520746723373001O472 0ustar
rootroot# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
BASH_ ENV=$HOME/.bashrc
USERNAME="root"

export USERNAME BASH_ENV PATH

— More —
```

참 고

압축파일의 확장자

보편화된 압축파일

.zip : DOS의 pkzip에서부터 사용된 압축형식으로서 대부분의 조작체계들과 압축프로그람에서 지원되고있는 가장 보편화된 압축형식이다.

.rar : 로씨야에서 처음 개발된것으로서 압축률이 높으면서도 제일 처음으로 분할압축을 지원하는 압축형식으로 높은 호평을 받았다.

.tar: UNIX명령인 tar를 리용하여 생성되는 파일형식으로서 원래는 테프기억매체의 예비보관을 위한 명령문을 의미하였다. 이 방식은 압축이 아니라 여러 파일들을 하나로 묶어놓은것이다. 그 리유는 예비보관한 파일이 잘못된 경우에도 압축하지 않은 파일은 일부라도 복구할수 있는 가능성이 그만큼 높기때문이다.

.tgz: UNIX에서 tar로 묶은 파일을 gzip로 압축한 파일형식이다.

.gz: UNIX용의 gzip에서 사용하는 압축형식으로서 원래는 OS/2에서 개발되였던 방식이다. 파일하나만을 압축할수 있으므로 주로 .tar와 함께 사용되며 이 경우에 .tar.gz 혹은 .tgz확장자를 사용한다.

.bz2: UNIX용의 bzip 2에서 사용하는 압축형식이다. 파일하나만을 압축할수 있으므로 주로 .tar와 함께 사용되며 이 경우에 .tar.bz2 형식의 확장자를 사용한다.

.arj: arj와 WinArj에서 사용하는 분할압축지원의 압축형식이다. UNIX 및 Linux에서도 사용할수 있다.

.Z: UNIX용의 compress, uncompress에서 사용하는 파일형식이다.

기타 압축파일

.ace: ACE에서 사용하는 분할형식지원의 압축형식이다. 성능이 매우 높다. 완벽한 다중기록권(Multi-volume)과 안정적인 압축으로 높은 평가를 받고있다.

.cab: 마이크로쏘프트가 지원하는 압축형식이다.

.ice: ice에서 사용하는 압축형식으로서 실지의 내용은 lha와 같다.

.jar: Java의 jar.exe에서 사용하는 압축형식이다. 내부적으로 zip압축알고리듬을 사용한다.

.lha, lzh: lha에서 사용하던 압축형식으로서 Lempel-Ziv알고리듬을 사용한다.

.mim: 인터네트에서 문서를 주고 받을 때에 사용하는 형식이다.

.pak: pak.exe에서 사용하던 압축형식이다.

znew

/usr/bin/znew

.gz로 변환

.z파일을 .gz로 변환하는 명령문이다.

형식 : znew[설정값][파일이름.z]

-f : .gz파일이 있어도 .z파일을 .gz파일로 다시 압축한다.
-t : 원본파일을 삭제하기전에 새로운 파일을 검사한다.
-v : 자세하게 보여준다.
-9 : 속도는 늦어지지만 압축률을 높인다.
-K : 원본파일이 .gz파일보다 작은 경우에 원래대로 유지한다.
-P : 파이프를 리용한다.

znew명령문은 compress로 압축된 .z파일을 gzip로 압축된 .gz파일로 변환하여준다.

```
kcc@linux:~
# compress test1
# ls
Desktop    command   query   test1.Z
Mail       expolit   study   troubleshoot
# znew test 1.Z
# ls
Desktop    command   query   test1.gz
Mail       expolit   study   troubleshoot
```

ztelnet

/usr/bin/ztelnet

원격접속

telnet의뢰기이지만 파일전송이 가능하다.

형식 : ztelnet[설정값]호스트명

-l user : 체계접속할 사용자를 지정한다.

telnet의 기능에 파일전송의 기능이 부가된것이다.

원래 telnet에는 파일전송기능이 없었지만 파일전송을 하는 rz/sz기능이 이후에 추가되였다.

[주의]

gau같은 PC통신프로그람을 사용하려고 하는데 내리적재가 올리적재되지 않는 경우가 있다. 이 경우에는 ztelnet가 체계에 설치되여있는가 먼저 확인해야 한다. 엄밀히 말하면 대면부인것이다.

```
kcc@linux:~

# ztelnet -l first home.kcc.net
Tryng home.kcc.net
connected to home.kcc.net
Escape character is '^]'.
Password:
```

home.kcc.net에 련결한 다음 내리적재할 파일을 선택하여 dn명령에 의해 ZMODEM으로의 내리적재를 시도한다.

어떤 이상한 수자의 조합으로 더 이상의 진행이 없게 되는데 이때 [ctrl+l]건입력으로 telnet를 탈퇴한 다음 rz명령을 입력하면 전자게시판(BBS)의 파일을 내리적재할수 있다.

이와 반대로 파일을 올리적재하는 경우에도 [Ctrl+J]건입력으로 telnet를 빠져나온 다음 sz명령으로 파일을 올리적재할수 있다.

【관련명령문】

telnet : 대표적인 원격체계접속의 명령으로서 국부호스트를 사용하는것처럼 쉘에서 여러가지 명령을 내릴수 있다.

ssh(Secure SHell) : telnet처럼 쉘에서 여러가지 명령을 사용하는데 telnet의 보안약점에 대처하여 사용자이름, 암호, 쉘의 명령문을 암호화하여 전송할수 있게 한다.

제 2 장
데몬(DAEMON)

UNIX나 Linux의 수많은 봉사가운데서 사용자들의 눈에 보이는것은 일부분에 지나지 않는다. 사용자들이 모르는 사이에 동작하며 봉사를 제공하고 체계의 조화로운 동작을 가능하게 하는것이 바로 데몬(Daemon)이다.

데몬은 사용자가 그 존재를 의식함이 없이도 주어진 상태에서 들어오는 요청을 하나씩 처리한다. 이 장에는 그러한 데몬들에 대하여 설명한다.

데몬(daemon)이란?

말단대화조종(terminal session)과 련결되여있지 않는 배경프로쎄스로서 이것은 주어진 시간이나 어떤 사건이 발생하였을 때에 지정된 봉사를 수행하게 된다. Linux배포판 RedHat 7.2판본인 경우, 데몬봉사실행파일스크립트들은 /etc/rc.d/init.d등록부아래에 존재한다. 다른 배포판들에서는 /etc/init.d등록부에 존재하는 경우도 있다.

아래의 화면출력은 다른 배포판과의 호환을 위해 /etc/init.d등록부를 rc.d/init.d에 쏘프트련결을 사용한것을 보여준다.

```
kcc@linux:~
# ls -al /etc/init.d
lrwxrwxrwx 1 root root 11 12월 14 2001 /etc/init.d -> rc.d/inrt.d
```

봉사기실행방식과 고급데몬

데몬실행방법에는 고급데몬(Super Damon)방식과 봉사기실행방식이 있다.

고급데몬은 《xinetd》라는 하나의 봉사기데몬에서 여러개의 봉사를 함께 관리하는 방법으로서 요청이 그리 많지 않은 봉사를 제공하는 경우에 유리하다. 이따금 들어오는 봉사요청을 하나의 xinetd데몬으로 관리를 한다면 봉사기의 자원을 그만큼 줄일수 있는것이다. 관련설정을 위해서는 /etc/xinetd등록부에 있는 파일들을 수정하여야 한다. RedHat 7.0이전판본에서는 inetd라는 이름으로 /etc/inetd.conf파일에서 설정을 하였다. 봉사기실행방식(Standalone)은 하나의 봉사기봉사에 매개의 데몬이 따로 실행되여 봉사한다. 이것은 봉사요청이 많은 경우에 사용하며 보통 웨브봉사, 우편봉사, 이름봉사 등에서 제공한다.

이 파일등록부에서는 많은 데몬관련스크립트들을 볼수 있다.

```
kcc@linux:~

# ls /etc/rc.d/init.d/
FreeWnn dhcpd  kadmin    mcserv  ptal-init  sendmanil  xfs
adsl   functions  kdcrotate  mysqld  pxe    single    xinetd
amd    gated      keytable  named  radvd    smb      ypbind
anacron   gpm     killall     netfs   random  snmpd yppasswdd
apmd      halt    kprop    network  rarpd   squid     ypserv
arpwatch  httpd   krb524    nfs   rawdevices  sshd    ypxfrd
atd       identd  krb5kdc  nfslock  reconfig  syslog
autofs   ipchains   kudzu   nscd    routed    tux
bcm5820  iptables   ldap    ntpd    rstatd    ups
bootparamd  irda  linuxconf  pcmcia  rusersd  vmware
crond      iscsi    lpd    portmap   rwalld   vncserver
dhcp-relay  kWnn  mars-nwe  proftpd rwhod  webmin
```

보통 데몬을 실행할 때에는 start명령을 사용하며 데몬을 정지시킬 때에는 stop를 실행한다.

```
kcc@linux:~

# /etc/rc.d/init.d/httpd start
Starting httpd ................[OK]
# /etc/rc.d/init.d/httpd stop
Stopping httpd ................[OK]
```

데몬을 다시 실행하기 위해서는 restart를 사용하기도 한다.

```
kcc@linux:~

# /etc/rc.d/init.d/httpd restart
Stopping httpd .............[OK]
Starting httpd .............[OK]
```

RedHat에서는 기동시에 실행데몬관리를 위해 ntsysv대면부를 제공하는데 이것은 chkconfig명령을 쉽게 안내방식으로 할수 있도록 도와주는 편의 프로그람이다.

```
# ntsysv
```

ntsysv는 기동시에 실행할 데몬을 공백건(space bar)으로 선택한 후에 확인단추를 선택하면 된다.

데몬이름	설　　명
adsl	ADSL접속스크립트
amd	장치적재(mount) 및 NFS호스트의 자동적재데몬
anacron	정지시간으로 실행되지 않는 cron job를 실행하는 스크립트
apmd	전지상태를 출력하고 syslog에 리력을 남긴다. 전지여유량이 적을 때에는 체계끝내기(shutdown)를 한다.
arpwatch	이써네트/IP주소조합을 추적유지하는 arpwatch명령데몬
atd	지정된 시간에 맞추어 실행되는 at명령실행일정을 위한 데몬
autofs	요구가 있는 즉시에 파일체계를 자동mount하도록 한다.
bootparamd	bootparamd봉사기는 이전의 Sun전용기가 Linux box로부터 망기동을 가능하게 한다. 현재는 대체로 bootp나 dhcp로 대신한다.
comsat	Comsat봉사기가 diff의뢰기로부터 새로운 우편의 도착을 알려주는데 사용된다. 고급데몬에서 관리.
crond	표준 UNIX프로그람으로 지정된 사용자프로그람을 주기적인 시간에 실행되도록 한다. vixie cron에는 보안과 설정관련의 세부적인 기능들이 추기되여있다.
dhcp-relay	DHCP relay agent를 위한 스크립트.
dhcpd	DHCP봉사기데몬
finger	finger요청에 봉사한다. finger는 원격사용자가 접속된 체계의 접속명과 마지막접속시간 등의 정보를 볼수 있게 허락하는 봉사로서 고급데몬방식으로 동작한다.
gated	경로지정데몬으로서 경로지정자료기지와 다중경로지정을 지원하는 통신규약모듈의 핵심봉사로 구성되여있는 모듈프로그람이다.
gpm	본문방식의 Linux응용프로그람을 지원하는 마우스설정을 허용한다.
httpd	Apache www봉사기데몬으로서 HTML과 CGI봉사를 한다.
identd	특별한 TCP접속을 하는 사용자의 동일성정의수단으로 제공된다.
imap	mutt, pine, fetchmail, Netscape Communicator 등의 IMAP의뢰기로 원격사용자가 그들의 우편을 접속할수 있도록 허락하는 IMAPS봉사.
imaps	Netscape Commnicator, fetcmail 등의 SSL을 지원하는 IMAP의뢰기로 원격사용자가 그들의 우편을 접속할수 있도록 허락하는 IMAPS 봉사.
ipchains	Kernel 2.2.x에서 지원하는 파케트추출방화벽을 자동화하는 봉사데몬.
ipop2	fetchmail과 같은 POP2의뢰기로 원격사용자가 그들의 우편을 접속할수 있도록 허락하는 POP2봉사.
pop3	Netscape communicator, mutt, fetchmail 등의 POP3의뢰기로 원격사용자가 그들의 우편에 접속할수 있도록 허락하는 POP3 봉사.

iptables	Kernel 2.4.x에서 지원하는 파케트추출방화벽을 자동화하는 봉사데몬
irda	IrDA지원을 실행/정지하는 하는 쉘스크립트
iscsi	iSCSI데몬
kWnn	Kserver를 실행/정지하는 쉘스크립트
kdcrotate	/etc/krb5.conf의 KDCs의 목록을 교체시키는 쉘스크립트.
keytable	/etc/sysconfig/keyboard의 설정된 정보를 적재한다. 이것은 kbdconfig편의프로그람으로 설정할수 있다.
kudzu	하드웨어검사 및 추가변경된 하드웨어를 찾는 스크립트.
ldap	LDAP(Lightweight Directory Access Procotol)는 표준등록부봉사를 실행/정지하도록 하는 쉘스크립트.
linuxconf	체계설정에 필요한 다양한 작업을 하는 linuxconf의 방책(Policy)을 적용시켜 준다.
lpd	인쇄기작업을 위해 요구되는 인쇄기데몬.
mysqld	mysql자료기지데몬.
named	호스트명을 IP주소로 변환하는 DNS(Domain Name Server)역할을 하는 named(BIND)데몬.
netfs	NFS(Network File System),SMB(Lan Manager/Window), NCP (NetWare)등의 모든 망파일을 mount/umount한다.
network	기동할 때에 모든 망대면부를 동작시키는 스크립트.
nfs	NFS(Network File System)봉사기데몬.
nfslock	NFS파일잠금봉사를 실행/정지하는 쉘스크립트.
pcmcia	학습장형콤퓨터에 망기판(NIC)이나 모뎀(MODEM)의 사용을 위해 PCMCIA를 지원한다.
pop3s	Fetchmail과 같은 SSL지원의 POP3의뢰기로 원격사용자가 우편접속할수 있도록 하는 POP3S봉사.
portmap	NFS, NIS, amd, mcserv 등 RPC를 사용하는 프로그람들을 위해 기본적으로 실행되여야 하는 중요한 데몬이다.
proftpd	ProFTPD는 단순하며 보안대책이 강구되고 설정을 쉽게 할수 있도록 확장된 FTP(File Transfer Procotocot)봉사기로 이것을 봉사하도록 하는 데몬이다.
random	체계에 필요한 란수를 발생/보관하는 스크립트.
rarpd	RARP(Reverse Address Resolution Procotol)을 요청하는 봉사데몬이다.
rawdevices	초기장치를 블로크형장치(하드디스크구획과 같은)로 할당하는 쉘스크립트로 /etc/sysconfig/rawdevices파일을 불러와 mapping한다.
reconfig	/etc/reconfigSys파일이 있는 경우에 이를 재설정하는 쉘스크립트.
rlogin rlogin	접속을 위한 봉사데몬.
routed	자동IP경로기를 위한 데몬.
rsh	rsh접속을 위한 봉사데몬.
rstatd	사용자가 망호스트의 성능을 추출가능하게 하는 데몬.

rsync	rsync는 rcp와 류사한 동작을 하는 프로그람으로서 rcp보다 다양한 설정값들이 있으며 보다 효률적으로 자료를 전송한다. 파일크기의 변화, 시간의 변화 등을 리용하여 호스트간동기화를 한다.
rusersd	특정사용자가 어느 망에 있는가를 찾아주는 봉사.
rwalld	원격사용자가 현재 체계에 접속한 사용자에게 통보문을 보낼수 있게 하여주는 봉사이다.
rwhod	원격사용자가 접속되여있는 사용자의 목록을 보기 위한 rwho명령을 위한 데몬.
sendmail	Sendmail우편봉사기데몬
sgi fam	FAM이 파일감시데몬으로 파일이 변경되였을 때 이 기록을 볼수 있다.
smb	Samba망파일봉사기데몬으로 봉사하기 위해 smbd와 nmbd데몬을 실행 및 중지시킨다.
snmpd	SNMP(Simple Network Management Procotol)봉사데몬으로 망상태를 감시할수 있다.
squid	HTTP, FTP, gopher 등과 같은 봉사의 caching속도를 높여주는 데몬이다.
sshd	Open SSH봉사데몬
swat	SWAT(Samba Web Admin Tool)는 Samba웨브관리자도구로 웨브페지를 통해 Samba를 관리할수 있게 한다.
syslog	체계의 다양한 사건을 리력정보로 보관하도록 하는 데몬으로서 반드시 선택할것을 권고한다.
talk	다른 체계사용자와의 실시간대화를 위한 데몬.
telnet	telnet봉사를 위한 데몬.
tftp	TFTP(Trival File Transfer Procotor)봉사데몬으로서 이것은 기동디스크가 없이 한 망내의 설정파일을 받아 기동할수 있게 한다.
webmin	체계관리를 위한 웨브도구인 webmin봉사데몬이다.
xfs	X서체봉사기로서 X Window에서 서체(font)를 쓰거나 봉사하도록 한다.
xinetd	접속통제수단, 확장접속호환, 봉사기수제한 등의 기능으로서 inetd를 지원하는 super daemon이다.
ypbind	NIS/YP의뢰기측에서 실행하는 봉사기로서 NIS를 사용하지 않는 체계에서는 실행하지 말아야 한다.
yppasswdd	NIS의뢰기사용자가 암호를 변경할수 있도록 해주는 NIS봉사기측의 데몬이다.
ypserv	표준 NIS/YP망통신규약봉사기로서 NIS의뢰기프로그람은 아니다.

atd

예약작업의 실행

at명령문으로 예약한 작업을 실행한다. at는 명령문과 스크립트파일을 특정시간에 실행할수 있도록 예약하는 기능을 한다.

형식 : atd[option]

-l : 동시작업이 실행되지 말아야 하는 부하제한상태로 되도록 한다.
번역시에 초기값으로 0.8이 적용된다. n개의 CPU를 가진 SMP체계에서는 n-1보다 큰 값을 설정해줄수도 있다.

at명령으로 수행한 작업을 atd데몬이 처리한다. 이것은 작업실행시간과 작업대기행렬을 유지관리한다. atd의 spool등록부는 보통 /var/spoll/at 등록부에 위치한다.

```
kcc@linux:~

# at 4am
warning: commands will be executed using (in order) a)
$SHELL b) login shell c) /bin/sh
at> echo "it’s 4am"
at> <EOT>
job 3 at 2004-06-25 04:00
Can’t open /var/run/atd.pid to signal atd. No atd running?
```

우의 통보문과 같이 atd데몬이 기동되지 않았다고 알려준다.
아래와 같이 다시 데몬을 실행하여 at명령을 실행해보자.

```
kcc@linux:~

# /etc/rc.d/init.d/atd start
starting atd....... [ OK ]
# at 4am
```

```
warning: commands will be executed using (in order) a)
$SHELL b) login shell c) /bin/sh
at> echo "it's 4am"
at> <EOT>
job 4 at 2004-06-25 04:00
```

우와 같이 새벽 4시에 4시라고 알려주는 echo명령이 한번 실행되는것을 볼수 있다. atq명령으로 예약된 작업을 보자. 3번작업과 4번작업이 있다.

```
kcc@linux:~
# atq
3    2004-06-25  04:00  a  root
4    2004-06-25  04:00  a  root
```

그중 3번작업을 삭제해보자.

```
# atrm 3
```

3번작업이 삭제된것을 확인할수 있다.

```
kcc@linux:~
# atq
4      2004-06-25 04:00 a root
```

이 작업은 /var/spool/at파일등록부에서 확인할수 있다.

```
kcc@linux:~
# ls -al /var/spool/at/
합계 20
drwx------  3 daemon  daemon  4096 6월 24 22:15 .
drwxr-xr-x 14  root     root    4096 3월 27 18:19  ..
-rw-------  1 daemon daemon  6     6월 24 22:14  .SEQ
-rwx------  1 root   root  2149 6월 24 22:09 a000040104a834
drwx------ 2 daemon daemon   4096  11월 13  2003  spool
```

crond /usr/bin/crond

예약작업의 실행

시간에 따라 등록된 명령을 정기적으로 수행하는 표준의 UNIX데몬

cron은 예비보관과 같이 주기적으로 실행하는 명령들을 자동적으로 처리하게 하는 도구로서 이를 위해서는 데몬을 실행한다.

```
kcc@linux:~
# /etc/rc.d/initd/crond start
Starting crond...........[OK]
```

cron기본설정파일은 RedHat의 경우 /etc/crontab에 있다

```
kcc@linux:~
# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

cron작업을 실행하기 위한 시간예약은 분, 시, 일, 월, 요일의 순서로 지정한다. 우의 설정파일내용가운데서 02 4 * * * root run-parts /etc/cron .daily의 의미를 보기로 하자.

02 4 * * * 는 분, 시, 일, 월, 요일의 순서로 시간과 날자를 분석해보면 매일 4시 2분에 /etc/cron.daily등록부의 내용을 실행한다는 의미이다.

그리고 22 4 * * 0의 의미는 매주 일요일의 4시 22분에 실행한다는 의미이다. 또한 42 4 1 * * 는 매월 1일 4시 42분에 실행한다는 의미이다.

이와 같이 매일 실행되는 예약작업은 /etc/cron.daily에, 매주 실행되는 예약작업은 /etc/cron.weekly에, 매월 실행되는 작업은 /etc/cron.monthly등록부에 실행한다는 의미이다.

시간과 날자에 대한 항목은 각각 다음과 같다.

분(Minutes)	0-59
시(Hour)	0-23
일(Day of month)	1-31
월(Month)	1-12 or Jan, Feb, Mar,...
요일(Day of week)	0-6(0은 Sunday)or Sun, Mon, Tue,...

최근의 Linux배포판들에서는 이전과 달리 매번 실행되는 일별, 주별, 월별로 등록부를 구분하여 관리하기 편리하도록 한것을 볼수 있다.

```
/etc/cron.hourly/
/etc/cron.daily/
/etc/cron.weekly/
```

dhcpd

동적 IP주소의 제공

bootp의 확장규약으로서 의뢰기들이 기동할 때에 자동적으로 동적인 IP주소와 망정보를 가질수 있게 하는 DHCP봉사기데몬이다.

DHCP(Dymmic Host Configuration Protocol)란 만일 DHCP의뢰기에서 IP주소를 요청하는 경우, 봉사기에서는 IP주소를 할당하는 방식이다.

이것은 초기적재의 통신규약을 확장한것으로서 할당된 IP주소에 임대기간을 설정할수 있는 점이 초기적재의 통신규약과 다르다.

BOOTP(Bootstrap Procotol)는 초기적재의 통신규약으로 망을 통한 체계기동이 가능하게 한다.

dhcp봉사기를 위한 설정은 /etc/dhcpd.conf이다. 만일 파일이 없는 경우에 생성하도록 한다.

```
kcc@linux:~

# cat /etc/dhcpd.conf
default-lease-time 600;
max-lease-time 7200
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option routers 192.168.0.1;
option  domain-name-servers 61.40.233.122;
option domain-name "ns.linuxroot.co.kp";

subnet 192.168.0.0 netmask 255.255.255.0 {
   range 192.168.0.10 192.168.0.30
   range 192.168.0.42 192.168.0.62;
}
```

default-lease-time	기본적으로 IP주소를 허가할 시간을 초단위로 지정한다.
Max-lease-time	IP주소의 최대시간을 초단위로 지정한다.
Options subnet-mask	의뢰기범위를 지정할 netmask를 설정한다.
Options broadcast -address	지정한 netmask의 범위에 맞게 IP대역의 제일 마지막 IP 주소를 지정한다.
Options router	Dhcp IP를 받을 의뢰기의 게트웨이를 지정한다.
Options domain -name-servers	이름봉사기를 지정한다.
Options domain-name	이름봉사기를 도메인이름으로 지정한다.

지정한 IP대역중 의뢰기에게 부여할 IP 범위를 지정할수 있다.

```
kcc@linux:~

subnet 192.168.0.0 netmask 255.255.255.0 {
   range 192.168.0.10 192.168.0.30
   range 192.168.0.42 192.168.0.62;
}
```

우에서 의뢰기에게 부여할 범위는 192.168.0.10~192.168.0.30, 192.168.0.42~192.168.0.62까지이다.

```
kcc@linux:~

host dhcp {
   hardware ethernet 00:50:BF:28:2A:4A;
   fixed-address 192.168.0.100;
}
```

망기판의 하드웨어주소(MAC Address)를 지정하여 이 기판에 고정적인 IP를 부여할수도 있다. /etc/dhcpd.conf파일을 설정한 다음 /etc/rc.d/init.d/dhcpd start명령으로 데몬을 실행하면 된다.

httpd

가장 널리 사용되는 Apache웨브봉사기

Apache의 기본설정파일은 /etc/httpd/conf/httpd.conf파일이다. 일반적으로 기본설정에서 특별히 수정을 하지 않아도 Apache웨브봉사기를 실행하는데는 문제가 없다.

```
kcc@linux:~

# cat /etc/httpd/conf/httpd.conf  |  more
# Based upon the NCSA server configuration fifes originally by
RobMcCool
#
# This is the main Apache server configuration file. It contains
# the configuration directives that give the server its
# instructions.
# See for detailed information about the directives
#
# Do NOT simply read the instructions in here without
# understanding what they do. They're here only as hints or
# reminders. If you are unsure consult the online docs.
# You have been warned.
#
# After this file is processed, the server will look for and
# process /etc/httpd/conf/srm.conf and then /etc/httpd
# /conf/access.conf unless you have overridden these with
# ResourceConfig and/or AccessConfig directives here.
#
# The confiuration directives are grouped into three basic
# sections:
# 1. Directives that control the operation of the Apache server
# process as a whole(the 'global environment').
# 2. Directives that define the parameters of the 'main' or
# 'default' server, which responds to requests that aren't
# handled by a virtual host.
--More--
```

체계의 기본 html뿌리등록부는 /var/www/html이다.

```
kcc@linux:~
# ls -al /var/www/html/
합계 28
drwxr-xr-x 5 admin admin 4096 2월 4 15:06 .
drwxr-xr-x 9 admin admin 4096 5월 2 15:11 ..
lrwxrwxrwx 1  admin  admin 24 12월 14 19:52 htdig -> ../../../usr/share/htdig
-rw-r—r— 1 admin admin 408 12월 3 09:34  index.html
drwxr-xr-x 9 admin admin 4096 2월 4 15:06 manual
drwxr-xr-x 2 admin admin 4096 12월 14 19:29 mrtg
-rw-r—r—  1 admin admin 1154 2월 1 15:46  poweredby.png
drwxr-xr-x 2 admin admin 4096 5 월 13 04:02 usage
```

우의 등록부에 index.html로 기본색인파일을 생성하면 된다.

apache데몬을 다시 시작하려면 다음과 같이 한다.

```
kcc@linux:~
# /etc/rc.d/init.d/httpd stop
Stopping httpd ....... [OK]
# /etc/rc.d/init.d/httpd start
Starting httpd ....... [OK]
```

웨브봉사기실행을 telnet의뢰기로 확인할수 있다. 기본적으로 웨브봉사기 포구는 Port 80이며 Connected통보문으로 apache데몬이 실행된것을 알수 있다.

```
kcc@linux:~
# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1)
Escape character is '^]'.
```

lpd

행인쇄기데몬

행인쇄기를 조종하는 데몬으로서 인쇄작업을 하는데 반드시 필요하다.

형식 : lpd[설정값][-p 포구]

-L logfile : 지정한 리력파일(logfile)에 오유나 오유수정통보문을 보존한다.

-F : 일반적으로는 LPD봉사기가 뒤단방식(background mode)으로 실행되지만 앞단방식(foreground mode)으로 실행하여 보다 오유수정이 쉽게 한다.

-V : 판본정보를 보여준다.

-p port : 기본포구인 515보다 지정한 포구(port)를 사용한다.

체계에서 국부령역에 직접 련결한 인쇄기나 원격망인쇄기, Samba인쇄기 등을 사용하려면 lpd데몬이 필요하다.

보통 RedHat에서 지원하는 인쇄기설정편의프로그람으로서는 본문환경에서 사용할수 있는 printconf-tui, GUI환경에서 사용할수 있는 printtconf-gui, printtool 등이 있다.

```
# /etc/rc.d/init.d/lpd start
```

인쇄기설정을 체계인쇄기환경에 맞게 설정한 다음에는 반드시 lpd데몬을 기동시켜야 한다.

인쇄기데몬을 기동시키기 위한 lpd스크립트는 다음과 같다.

아래의 스크립트는 printconf-backend명령의 실행권한을 확인하고 /etc/printcap파일이 존재하는가를 확인한 다음 lpd데몬을 실행하게 되여있다.

```
kcc@linux:~
start ( ) {
    echo -n $"Starting $prog: "
    # Is this a printconf system?
    if [[ -x /usr/sbin/printconf-backend ]]; then
         # run printconf-backend to rebuild printcap  and spools
         if ! /usr/sbin/printconf-backend ; then
            # If the backend fails, we dont start no printers defined
            echo -n $"No Printers Defined"
            echo_success
            echo
            return  0
         fi
    fi
    if ! [ -e/etc/printcap ] ; then
       echo_failure
       echo
       return 1
    fi
    # run checkpc to fix whatever lpd would complain about
#  /usr/sbin/checkpc -f
   # start  daemon
   daemon /usr/sbin/lpd
   RETVAL=$?
   echo
   [ $RETVAL=0] && touch /var/lock/subsys/lpd
   return  $RETVAL
}
```

기본적으로 인쇄기설정등록부는 /etc/printcap이며 spool파일등록부는 /var/spool/lpd이다.

```
kcc@linux:~

# cat /etc/printcap
        :sh:\
        :ml=0:\
        :mx=0:\
        :sd=/var/spool/lpd/lp:\
        :rm=61.40.233.14:\
        :lpd_bounce=true:\
        :if=/usr/share/printconf/util/mf_wrapper:
```

lp=	인쇄기장치의 파일명이다.
mx=	인쇄기작업파일의 최대크기를 나타낸다.
sd=	spool등록부를 지정한다. /var/spool/lpb등록부의 인쇄기장치명이 spool등록부로 지정되였다.
rm=	원격인쇄기의 호스트명

이외에 국부환경인쇄기를 원격환경인쇄기봉사기로 사용하는 경우 원격으로 인쇄할수 있도록 허용할 호스트를 지정할수 있는데 /etc/hosts.equiv나 /etc/hosts.lpd에 호스트를 추가한다.

【관련명령문】

lpc : 행인쇄기를 조종한다.
lpq : 기본인쇄기대기렬의 내용을 보여준다.
lpr : 파일내용을 출력한다.
lprm : 인쇄기대기렬의 작업을 삭제한다.
lpstat : LP인쇄기의 정보를 보여준다.

named

이름봉사기데몬

이름봉사기에 봉사하는 bind봉사기의 데몬이다.

이름봉사기는 도메인이름을 IP로 넘긴 자료를 가지고있으면서 도메인에 대한 질문이 들어오는 경우에 해당하는 IP를 알려주게 된다. 례를 들어 망설정에서 이름봉사기를 지정하지 않는 경우나 지정한 이름봉사기에 이상이 있는 경우에 웨브열람기나 다른 인터네트봉사에서 URL로는 접근이 불가능하게 되며 해당되는 봉사기의 IP로만 접근할수 있다.

Linux에서 이름봉사기로서 Bind(Barkeley Internet Name Domain)가 사용된다. 아래의것은 이름봉사기설정에 필요한 파일들의 목록이다.

- /etc/named.conf: 기동파일로서 첫번째 설정파일.
- /var/named/named.ca: 경로이름봉사기에 대한 정보가 있는 자료기지파일(cache파일)
- /var/named/named.local: localhost에 대한 설정파일
- /var/named/forward.zone: Public Domain에 대한 Forward zone파일
- /Var/named/reverse.zone: Inverse Domain에 대한 Reverse zone파일

1. named.conf의 설정

named.conf파일은 이름봉사기를 위한 환경설정파일로서 기본적으로 cache파일과 localhost에 대한 설정이 되여있다. 여기에 신청한 도메인의 Forward zone파일과 Reverse파일에 대한 설정을 추가해야 한다.

아래에서는 linuxroot.co.kp도메인에 대한 Forward zond파일로 linuxroot.zone을 설정하였으며 reverse zone파일로 linuxroot.rev를 설정하였다.

kcc@linux:~

```
# cat /etc/named.conf
// generated by named-bootconf.pl
options {
directory "/var/named";
/*
* If there is a firewall between you and nameservers you want
* to talk to, you might need to uncomment the query-source
* directive below. Previous versions of BIND always asked
* questions using port 53, but BIND 8.1 uses an unprivileged
* port by default.
*/
// query-source address * port 53;
};

//
// a caching only nameserver config
//
zone "." IN {
type hint;
file "named.ca";
};

zone "localhost" IN {
type master;
file "localhost.zone";
allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
type master;
file "named.local";
allow-update { none; };
};
zone  "linuxroot.co.kp" IN {
type master;
```

```
file "linuxroot.zone";
allow-update { none; };
};

zone "233.40.61.in-addr.arpa" IN {
type master;
file "linuxroot.rev";
allow-update { none; };
};
[root@ns root]#
```

linuxroot.co.kp도메인에 대한 named.conf파일의 설정을 우에서 보여주었다.

우의 사항과 같이 Forward zone파일과 Reverse zone파일을 지정한다.

파일이름은 신청한 도메인이름을 따서 linuxroot.zone과 linuxtoot.rev 파일로 지정한다. zone파일명은 지정된것이 없으며 named.conf파일에서 지정한 파일을 사용하게 된다.

우의 설정에서 주의할것은 233.40.61.in-addr.arpa는 61.40.233.122 호스트의 Reverse zone설정으로 IP를 거꾸로 입력해야 하며 마지막 8bit는 입력하지 않는다는것이다.

2. linuxroot.zone의 생성

named.conf파일에서 신청한 도메인인 linuxroot.co.kp에 대한 Forward zone파일과 Reverser zone파일은 각각 linuxroot.zone과 linurroot.rev로 설정하였다. 매 파일들은 /var/named등록부에 파일들을 생성한다.

/var/named에로 이동하여 파일들을 확인해보자.

기본적으로 named.ca, localhost.zone, named.local파일은 생성되여있다. named.local파일을 linuxtoot.zone파일로 복사한다.

```
kcc@linux:~

# cd /var/named
# ls
localhost.zone named.ca named.local
# cp named.local linuxroot.zone
```

형식에 맞게 linuxroot.zone파일을 편집한다.

```
kcc@linux:~

# vi /var/named/linuxroot.zone
$TTL 86400
@ IN SOA ns.linuxroot.co.kp. root.ns.linuxroot.co.kp. (
2004031200 ; Serial
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400 ) ; Minimum
IN NS ns.linuxroot.co.kp.
IN MX  10 mail.linuxroot.co.kp.

ns IN A 61.40.233.122
linuxroot.co.kp. IN A 61.40.233.122
mail IN A 61.40.233.122
www IN A 61.40.233.122
ftp IN A 61.40.233.122
```

[주의]

편집할 때 도메인이름다음에는 반드시 점《.》을 찍어야 한다.

[주의]

관리자우편주소는 root@ns.linuxroot.co.kp가 아니라 root.ns.linuxroot.co.kp로 입력한다.

형 식

;	주해
TTL(Time to Live)짧게 입력한다	cache활성시간으로서 봉사기정보를 자주 변경하는 경우에 본다
@	Origin을 의미하는 특수문자이다.
IN(Internet)	망주소집합
SOA(Start of Authority)	이름봉사기관리자의 우편주소: 해당 도메인의 모든 자원에 대한 권리를 알려주는 레코드.
NS(Name Server)	이름봉사기를 알린다.
MX(Mail Exchanger)	지정하는 우편봉사기가 2개이상일 때 우선순위를 지정한다.
A(Address)	도메인을 IP로 넘기는 가장 중요한 레코드이다.

[주의]

MX가 낮게 설정된 봉사기가 먼저 우편을 받고 실패할 경우 다음 봉사기가 우편을 받는다.

그 외에도 CNAME와 HINFO기록이 있다.

CNAME(Canonical NAME)는 하나의 도메인으로 alias기능을 사용할수 있으며 HINFO(Host INFOration)는 이름봉사기의 설명을 나타낸다.

3. linuxroot.rev의 생성

forward zone파일인 linuxroot.zone을 생성하였다면 reverse zone파일인 linuxroot.rev파일만 생성하면 된다. 형식은 거의 같으며 A기록을 PTR레코드로 지정하면 된다.

linuxroot.rev파일을 생성하기 위해 named.local파일을 linuxtoot.rev 파일로 복사한다.

```
kcc@linux:~

# cd /var/named
# ls
linuxroot.zone localhost.zone named.ca named.local
# cp named.local linuxroot.rev
```

linuxroot.rev파일을 다음과 같이 편집한다.

```
kcc@linux:~

# vi /var/named/linuxroot.rev $TTL 86400
@ IN SOA ns.linuxroot.co.kp. root.ns.linuxroot.co.kp. (
2004031200 : Serial
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400 ) ; Minimum
IN NS ns.linuxroot.co.kp.

122 IN PTR ns.linuxroot.co.kp.
```

4. 형 식

reserve zone파일은 forward zone파일을 편집하였다면 쉽게 알수 있다. 기본적인 형식은 같으며 IP를 도메인으로 넘길수 있도록 PTR(Pointer)기록을 사용하여 도메인에 대한 IP를 지정하면 된다.

5. 그외의 설정

이름봉사기를 설정하는 봉사기의 /etc/resolve.conf파일을 이름봉사기 설정에 맞게 수정한다. /etc/hosts파일과 /etc/host.conf파일은 참고로 보기만 하자. /etc/resolv.conf파일은 봉사기의 이름봉사기를 지정하는것으로서 여러개의 이름봉사기를 등록할수 있으며 입력된 순서대로 지정된 이름봉사기에 문의하게 된다. 첫번째로 등록될 이름봉사기는 NIC이나 KPNIC에 등록된 1차이름봉사기로 ns.linuxroot.com을 설정하는 봉사기의 IP를 아래와 같이 입력한다.

```
kcc@linux:~

# cat /etc/resolv.conf
nameserver 61.40.233.122
nameserver 168.126.63.1
```

/etc/hosts파일은 간단한 이름봉사기의 역할을 하는것으로서 자주 사용하는 도메인에 대한 alias를 지정할수 있다.

```
kcc@linux:~

# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 localhost.localdomain localhost
61.40.233.122 ns.linuxroot.co.kp.ns
```

/etc/host.conf파일은 아래와 같이 hosts와 bind에 대한 참조순서를 결정한다.

```
kcc@linux:~

# cat /etc/host.conf
order hosts,bind
```

network

망구동스크립트

기동할 때에 모든 망대면부를 동작시키는 스크립트이다.

1. IP Address의 설정

/etc/sysconfig/network-scripts/ifcfg-eth0파일은 eth0대면부의 IP Address, Netmask, Netwark Adress, Broadcast Address, 기동할 때 인식의 여부 등을 설정한다.

```
kcc@linux:~

# vi /etc/sysconfig/network-scripts/ifcfg-eth 0
DEVICE=eth 0
IPADDR=61.40.233.122
NETMASK=255.255.255.0
NETWORK=61.40.233.32
BROADCAST61.40.233.255
ONBOOT=yes.
```

2. Gateway의 설정

/etc/sysconfig/network파일은 ip forward enable여부의 설정과 Hostname, Domainname, Gateway 등을 설정하는 파일이다.

```
kcc@linux:~

# vi /etc/sysconfig/network
NEWORKING=yes
FORWARD_IPV4=false
HOSTNAME=ns.linuxroot.co.kp
DOMAINNAME=
GATEWAY=61.40.233.1
GATEWAYDEV=eth0
```

3. 이름봉사기의 설정

/etc/resolv.conf파일은 이름봉사기주소를 지정하는 파일이다.

```
kcc@linux:~
# vi /etc/resolv.conf
nameserver 61.40.233.122
```

4. network데몬

매 본문파일을 망에 맞게 수정을 진행한 후에 network데몬을 다시 시작해주면 망설정이 변경된다.

```
kcc@linux:~
# /etc/rc.d/init.d/network restart
대면부 eth0을 완료함 :                        [확인]
대면부 eth1을 완료함 :                        [확인]
IPv4파케트 forwarding을 완료하고있습니다 :  [확인]
망매개변수를 설정하고있습니다 :              [확인]
lO대면부를 시작하는 중입니다 :                [확인]
eth0대면부를 활성화중입니다 :                 [확인]
eth1 대면부를 활성화중입니다 :                [확인]
```

nfs

NFS봉사기데몬

Linux는 NFS(망파일체계)통신규약을 리용한 파일체계공유에 있어서 의뢰기로서뿐만아니라 봉사기로도 역할을 할수 있다. NFS는 UNIX체계사이에 파일체계의 마운트(mount: 적재, 설치)를 지원하는 사실상의 표준이다. 그러나 체계가 NFS봉사를 제공하는것은 보안상의 문제를 발생시킬수 있다. 그러므로 개인들은 이 봉사를 될수록이면 사용금지해야 한다.

NFS공유의 설정을 위해서는 /etc/exports파일을 수정해주면 된다.

```
kcc@linux:~
# vi /etc/exports
/hdc1 nfs.linuxroot.co.kp(ro)
/movie 61.40.233.123(ro)
```

첫번째 행에서는 nfs.linuxroot.co.kp호스트가 /hdc1등록부를 읽기전용으로 마운트하며 두번째 행에서는 61.40.233.123호스트가 /movie등록부를 읽기전용으로 적재한다. 설정후에는 nfs데몬을 재실행하면 된다.

```
kcc@linux:~
# /etc/rc.d/init.d/nfs start
starting nfs ....... [OK]
```

linuxtoot.co.kp호스트에 NFS봉사기설정을 하였다면 의뢰기에서는 다음과 같이 마운트할수 있다.

```
# mount -t nfs linuxroot.co.kp:/hdc1 /mnt/data
```

mount명령으로 파일체계형식을 nfs로 지정하며 linuxroot.go.kp호스트의 /hdc1파일등록부를 /mnt/data등록부로 마운트한다.

proftpd

ftp데몬

ftp봉사를 위한 proftp봉사기데몬

FTP(File Transfer Protocol)

의뢰기와 봉사기사이의 파일을 전송하기 위한 통신규약이다.

proftp봉사기관련명령문들은 다음과 같다.

- /etc/ftpusers: ftp접속을 허가하지 않는 식별이름목록
- /etc/pam.d/ftp: ftp보안관련설정파일
- /etc/proftpd.conf or /etc/proftpd/proftpd.conf: proftpd기본설정파일
- /etc/rc.d/init.d/proftpd: proftpd standalone데몬
- /var/ftp/incoming: anonymous ftp파일의 올리적재등록부
- /var/ftp/pub: anonymous ftp파일의 내리적재등록부
- /usr/bin/ftpcount: 매 proftpd server설정에 접속되여있는 사용자수를 출력
- /usr/bin/ftpwho: 매 ftp user들의 현재 process정보를 출력
- /usr/bin/ftpls: ftp등록부목록을 출력
- /usr/bin/ftpcopy: 목적지ftp로부터 파일내리적재
- /var/log/xferlog: ftp리력파일

1) /etc/ftpusers

ftpusers파일에 추가된 user들은 ftp접속을 하지 못하게 한다. 기본적으로 체계 user들이 포함되여있다.

```
kcc@linux:~
# cat/etc/ftpusers
root
bin
daemon
adm
lp
sync
---중간 생략---
anonymous
```

ftpusers에 anonymous식별이름(account)이 추가되여있어 이 ftp봉사기에서는 닉명사용자의 접근을 허용하고있지 않다.

2) /etc/pam.d/ftp

PAM(Pluggable Auhentication Modules)은 체계관리자가 응용프로그람들의 사용자를 인증하는 방법을 선택할수 있게 하는 공유서고묶음이다.

```
kcc@linux:~
# cat /etc/pam.d/ftp
# %PAM-1.0
auth required /lib/security/pam_listfile.so item=user sense=deny file
=/etc/ ftpusers onerr=succeed
auth required /lib/security/pam_pwdb.so shadow nullok
# This is disabled because anonymous logins will fail otherwise,
# unless you give the 'ftp' user a valid shell, or/bin/false and add
# /bin/false to /etc/shells
# auth required /lib/security/pam_shells.so
account required /lib/security/pam_pwdb.so
session required /lib/secumy/pam_pwdb.so
```

우에서 보는바와 같이 file=/etc/proftpd/conf/ftpusers라고 설정되여있으므로 fitpusers파일에 있는 사용자들은 sense=deny에 의해 거부되여있다
.

3) /etc/proftpd.conf

```
kcc@linux:~

# cat /etc/proftpd.conf
ServerName "Proftpd FTP Server"
ServerType standalone
DefaultServer on
Port 21
Umask 022
MaxInstances 30
User nobody
Group nobody
UseReverseDNS off
IdentLookups off
AuthPAMAuthoritative on
RootLogin off
DenyFilter \*.*/
DeferWelcome on
TimesGMT off

TimeoutIdle 0
TimeoutNoTransfer 0
TimeoutLogin 300

DisplayLogin /etc/proftpd/welcome.msg
DisplayFirstChdir .message

<Directory /*>
AllowOverwrite on
</Directory>
```

ServerName	ftp봉사기이름
ServerType	ftp데몬이 고급데몬인 inetd에 의해 구동될것인가, standalone방식으로 ftp를 구동할것인가를 선택
DetailtServer	기본 ftp로 설정
Port	ftp사용포구를 지정. 기본적으로 21번을 사용
Umask	기본적으로 022를 사용
MaxInstances	자식프로쎄스의 최대수를 지정
User, Group	사용자와 집단을 지정

UseReverseDNS	접속한 IP주소에 대한 Reverselookup허용의 여부
IdentLookups	일반적으로 의뢰기가 ident protocol을 remote usename에 대한 허용여부
AuthPAM Authoritative	PAM이 인증에 있어서 최종단계의 권한할당
RootLogin	뿌리의 ftp접속허용여부
DenyFilter	어떤 명령과도 부합되지 않는 정규표현식을 지정. 이것은 proftpd로 오는 어떤 명령문조합을 막는데 아주 유용하다.
DeferWelcome	의뢰기가 인증에 성공할 때까지 봉사기에 대한 정보를 전송 지연.
TimesGMT	GMT시각의 사용여부
RateReadBPS	주어진 설정부분에서 초당 전송바이트수를 지정
RateReadFreeBytes	대역폭의 제한없이 전송되는 바이트수
RateReadHardBPS	RateReadFreeBytes에 지정된 량만큼 파일이 전송되면 RateReadBPS에 정해진것만큼 대역폭을 사용
TimeoutIdle	접속자가 입력없이 련결을 유지할수 있는 초단위의 최대시간
TimeoutNoTransfer	접속자가 파일을 주고 받거나 등록부목록을 받는것과 같은 련결행위에 대한 입력없이 유지할수 있는 초단위의 최대시간
TimeoutLogin	의뢰기가가 인증을 유지할수 있는 시간을 초단위로 지정.
Displaylogin	Login때에 출력되는 통보문을 지정
DisplayFirstChdir	Cd명령에 의해 처음으로 등록부를 이동하였을 때에 나타나는 통보문을 지정.
llowOverwrite on	새로 생성될 파일로 덧쓰기를 허용

4) /etc/rc.d/init.d/proftpd

기본적인 설정을 모두 하였다면 설정을 적용하기 위해 아래와 같이 데몬을 다시 시작한다.

```
kcc@linux:~
# /etc/rc.d/init.d/proftpd stop
Shutting down proftpd: Suspending NOW [OK]
# /etc/rc.d/init.d/proftpd start
Allowing sessions again [OK]
```

sendmail

우편봉사기데몬

대표적인 우편전송봉사기이다.

먼저 우편과 관련된 기본적인 용어를 보기로 하자.

- SMTP(Simple Mail Transfer Protocol): 우편을 보낼 때에 사용되는 통신규약-포구 25번
- POP 3(Post Office Protocol): 봉사기에 있는 우편을 의뢰기로 가져올 때에 사용되는 통신규약-포구 110번
- IMAP(Internet Message Access Protocol): POP3이 보낸 우편에 대해 봉사기에 더 이상 남아있지 않지만 IMAP로 접속하여 우편을 읽으면 우편봉사기에는 우편이 계속 존재한다.
- 포구 143번: 례) 뉴스집단(NewsGroup)
- MUA(Mail User Agent): 우편을 보내기 위해 사용되는 프로그람
  례) Outlook, Netscape Messenger
- MTA(Mail Transfer Agent): 우편을 전송받아 이것을 외부에로 전송하는 프로그람
  례) sendmail, Qmail
- MDA(Mail Delivery Agent): 전송받은 우편을 해당 사용자에게 전달하는 역할로서 동일한 우편봉사기에서 서로 우편을 주고받는 경우

sendmail설정의 기본파일은 /etc/sendmail.cf파일이며 그외의 설정파일들은 /etc/mail/등록부에 모두 모여있다.

```
kcc@linux:~
# ls -al /etc/sendmail.cf
-rw-r—r—1 root root 46298 12월 3 2003 /etc/sendmail.cf
합계 108
# ls -al /etc/mail
drwxr-xr-x 2 root root 4096 5월 20 16:40 .
```

```
drwxr-xr-x 68 root root 8192 6월 10 13:47..
-rw-r—r— 1 root root 611 12월 3 2003 Makefile
-rw-r—r— 1 root root 364 3월 27 18:22 access
-rw-r—r— 1 root root 12288 5월 20 16:40 access.db
-rw-r—r— 1 root root 0 12월 3 2003 domaintable
-rw-r—r— 1 root root 12288 5월 20 16:40 domaintable.db
-r—r—r— 1 bin bin 5555 3월 11 15:54 helpfile
-rw-r—r— 1 root root 64 12월 3 2003 local-host-names
-rw-r—r— 1 root root 0 12월 3 2003 mailertable
-rw-r—r— 1 root root 12288 5월 20 16:41 mailertable.db
-rw-r—r— 1 root root 2475 12월 3 2003 Sendman.mc
-rw-r—r— 1 root bin 628 5월 14 04:02 statistics
-rw-r—r— 1 root root 127 12월 3 2003 trusted-users
-rw-r—r— 1 root root 0 12월 3 2003 virtusertable
-rw-r—r— 1 root root 12288 5 월 20 16:40 virtusertable.db
```

1. sendmail.cf파일의 설정

sendmail.cf파일은 설정파일자체가 암호화와 같은 설정을 취하고있으며 전문적인 기능을 사용하지 않는 경우라면 기본적인 설정을 그대로 두어 변경하지 않는다. sendmail.cf파일의 Cwlocalhost는 우편봉사기의 수신을 받기 위한 도메인을 지정하는 부분으로서 /etc/mail/local-host-names와 같은 역할을 한다.

2. /etc/mail/local-host-names설정

이름봉사기를 운영하며 가상도메인봉사를 하는 경우 local-host-names에 봉사하고있는 도메인을 추가적으로 기입하여준다.

3. /etc/mail/access설정

특정호스트나 도메인의 우편봉사기에 대한 접근을 설정한다. sendmail server에서 access할 호스트에 대해 우편중계(Relay)의 허용유무를 결정한다. 우편봉사기를 사용할 기관내망대역(보통 C class)을 추가한다.

```
kcc@linux:~
61.40.233.   RELAY
# vi /etc/mail/local-host-names
lacalhost.localdomain RELAY
localhost RELAY
127.0.0.1 RELAY
61.40.233. RELAY
```

⚠ **경 고**

특정 도메인이나 호스트에 대한 접근을 통제할수 있다.

linuxroot.co.kp : 특정 도메인에 대해서 접근통제

mail.lnxroot.co.kp : 특정 호스트에 대해서 접근통제

61.40.233.122 : 특정 IP주소에 대한 접근통제

admin@ : 특정 사용자에 대한 접근통제

⚠ **경 고**

접근통제설정값

OK : 지정한 호스트나 도메인에 대한 허용

RELAY : RELAY허용

REJECT : 수신 및 발신의 거부(반송)

DISCARD : 받은 우편의 폐기

501 message : 특정 사용자 및 주소에 대한 통보문반송접근거부

550 message : 특정 도메인에 대한 통보문의 반송

570 message :특정 도메인 또는 특정우편에 대한 경고통보문 및 우편을 거부

Spam우편봉사기의 악용을 방지하기 위해 makemap hash /etc/mail/access </etc/mail/access명령으로 access자료기지를 갱신한다.

기본설정이 되였다면 /etc/rc.d/init.d/sendmail start로 sendmail데몬을 실행한다.

```
# makemap hash /etc/mail/access < /etc/mail/access
```

```
kcc@linux:~

# /etc/rc.d/init.d/sendmail stop
Stopping sendmail ............ [OK]
# /etc/rc.d/init.d/sendmail start
Starting sendmail .............. [OK]
```

smb

Samba봉사데몬

Samba설정은 /etc/samba/smb.conf파일에서 한다.

```
kcc@linux:~
#------------------------Global Settings
[global]
workcroup=WORKGROUP-> Windows의 작업집단과 꼭같은 집단을
지정한다.
Server string=File Server -> Linux콤퓨터에 대한 설명
; hosts allow=192.168.1. 192.168.2. 127. -> 접근허용하는 망범위
printcap name=/etc/printcap
load printers=yes
printing=lprng
guest account=pcguest
log File=/var/log/samba/%m.log
max log size=0
security=share
encrypt passwords=yes
smb passwd file=/etc/samba/smbpasswd
socket options=TCP_NODELAY SO_RCVBUF=8192
SO_ SNDBUF=8192
dns proxy=no
#------------------------ Share Definitions
[homes]
comment=Home Directories
browseable=no
writable=yes
[printers]
camment=All Printers
path=/var/spool/samba
browseable=no
# Set public=yes to allow  user 'guest account' to  print
guest ok=no
printable=yes
[public]                  //공유할 파일등록부이름
comment=Public Stuff      //설명
path=/home/samba          //samba 공유등록부
```

```
public=yes
writable=yes
printable=no
[movie]
camment=movie's directory
path=/home/movie
valid users=jkwoo
public=no
writable=yes
printable=no
```

웨브열람기를 통하여 Samba봉사기를 쉽게 설정할수 있는 SWAT라는 프로그람이 있다. 이 프로그람을 사용하려면 /etc/xinetd.d/swat파일을 다음과 같이 수정한다.

```
kcc@linux:~
# cat /etc/xinetd.d/swat
# default: off
service swat
{
        disable         =no
        port            =901
        socket_type     =etream
        wait            =no
        only_from       =127.0.0.1
        user            =root
        server          =/usr/sbin/swat
        log_on_failure  += USERID
}
```

disable=no로 수정한 다음 /etc/rc.d/init.d/xinetd데몬을 재실행한다.

swat프로그람을 열람기에서 실행하려면 http://localhost:901을 주소창에 입력한다.

sshd

OpenSSH봉사기데몬

SSH는 두 호스트간의 통신암호화와 사용자인증을 위해 공개열쇠암호방식을 사용한다. 대화가로채기(Session Hijacking)와 DNS에 대한 부정접근시도를 방지해주면서 원격호스트에 접속하거나 호스트들사이에 자료를 복사하기 위해 사용될수 있다. 이로부터 보안을 위해 telnet봉사대신에 ssh봉사를 리용할것을 권고한다.

ssh봉사기설정파일은 /etc/ssh/sshd_config파일이다.

```
kcc@linux:~

# ls -al /etc/ssh/
합계 140
drwxr-xr-x 2 root   root   4096   6월 25   15:04 .
drwxr-xr-x 69 root  root   8192   6월 25   14:53 ..
-rw-r—r—   1 root   root   5      12월 14  2003 denyuser
-rw———     1 root   root   88039  12월 5   2003 moduli
-rw-r—r—   1 root   root   1050   12월 5   2003 ssh_config
-rw———     1 root   root   668    12월 14  2003 ssh_host_dsa_key
-rw-r—r—   1 root   root   590    12월 14 2001 ssh_host_dsa_key.pub
-rw———     1 root   root   515    12월 14  2003 ssh_host_key
-rw-r—r—   1 root   root   319    12월 14 2003 ssh_host_key.pub
-rw———     1 root   root   887    12월 14  2003 ssh_host_rsa_key
-rw-r—r—   1 root   root   210    12월 14 2003 ssh_host_rsa_key.pub
-rw———     1 root   root   2026   12 월 5 2003 sshd_config
```

기본설정에서 변경할 내용은 없다.

denyuser파일은 ssh봉사를 제외시킬 식별이름을 추가한다.

ssh봉사를 위해 /etc/rc.d/initd/sshd start로 데몬을 실행시킨다.

ssh의리기사용법은 다음과 같다.

접속할 식별이름은 admin이고 접속할 ssh봉사기가 localhost이라고 가정하자. 이 식별이름이나 그 @접속호스트에로 접근할수 있다.

```
# ssh -l admin linuxroot.co.kp
```

다음과 같은 방법으로도 사용할수 있다.

```
# ssh admin@61.40.233.122: movie ./
```

이 명령을 실행하면 다음과 같은 통보문이 출력되는것을 볼수 있다.

```
kcc@linux:~
The authenticity of host 'linuxroot.co.kp (61.40.23.122) can't be established.
RSA key fingerprint is fa:42:1b:b3:64:1b:23;1c:56:d5:
70:69:e4:a6:86:56.
Are you sure you want to continue connecting (yes/no)?
```

우의 통보문은 ssh로 해당봉사기에 처음 접속할 때에만 나오며 접속할 봉사기의 호스트열쇠가 ~/ssh/known_hosts(ssh2의 경우 known_hosts2) 파일에 보관된다. yes를 입력하여 호스트열쇠를 받게 되면 암호를 입력하여 접속할수 있게 된다.

```
kcc@linux:~
Warning : Permanently added 'localhost' (RSA) to the list
of known hosts.
root@localhost's password:
```

참　　고

ssh봉사기에 접근함이 없이 파일복사하기

ssh명령으로 접속하지 않아도 s명령을 사용하기전에 인증확인을 거쳐 봉사기의 파일에 접근할수 있다. scp명령을 사용하여 봉사기의 파일을 복사할수 있다.

scp명령으로 61.40.233.122의 ssh봉사기의 /movie등록부내의 모든 파일을 현재의 국부등록부에로 복사할수 있다.

syslog

체계에 발생하는 다양한 리력들을 기록하는 봉사

체계리력에는 핵심부(kernel)와 여러가지 체계관련의 오유, 통보문들을 가지고있다. 체계의 오유나 정보파악들에 아주 유용하고 중요한 정보, 리력정보를 /var/log등록부의 파일에 보관하게 되는데 이 역할이 syslog이다.

원격호스트에 리력을 기록하기

리력기록은 체계의 정보를 위한 중요한것이다. 체계가 해킹을 당하는 경우에도 그를 다른 체계에 기록하여 리력정보를 삭제하지 못하도록 설정해 보자.

1. 리력기록을 보내는 봉사기에서의 설정

```
kcc@linux:~

# cat /etc/syslog.conf
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*             /dev/console

# Log anytning (except mail) of level info higher.
# Don't log private  authentication messages!
*.info;mail.none;news.none;authpriv.none;cron.none
/var/log/messages

# The authpriv file has resricted access.
authpriv.*              /var/log/secure
# Log all the mail messages in one place.
mail.*                  /var/log/mai;llog
# Log cron stutff
cron.*                  /var/log/cron
# Everybody gets emergency messages
*.emerg                         *
# Save news errors of level crit and higher in a special file.
"/etc/syslog.conf" 33L, 937C              1.1     Top
```

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                          /dev/console
# Log  anything (except mail) of level info or higher.
# Don' log private authentication messages!
*.info;mail.none;news.none;authpriv.none;cron.none
/var/log/messages
# The authpriv file has restricted access.
authpriv.*                                        /var/log/secure
# Log all the mail messages in one place.
mail.*                                           /var/log/maillog
# Log cron stuff
cron.*                                           /var/log/cron
# Everybody gets emergency messges
*.emerg                                         *
# Save news errors of level crit and higher in a special file.
uucp,news.crit                                  /var/log/spooler
# Save boot messages also to boot.log
local7.*                                        /varlog/boot.log
#
# INN
#
news.=crit                                      /var/log/news/news.crit
news.=err                                       /var/log/news/news.err
news.notice                                     /var/log/news/news./notioe
```

모든 리력기록을 logserver에 보내기 위해서는 다음과 같이 한다.

```
*.*             @logserver
```

mail에 관련된 리력을 logserver로 보내기 위해서는 다음과 같이 한다.

```
mail*          @logserver
```

2. 리력기록을 받는 봉사기에서 필요한 설정

/etc/rc.d/init.d/syslog데몬스크립트에서 《SYSLOGD_OPTIONS》변수를 《-m o》로 지정하고 《daemon syslogd》명령문에 -r -h설정값을 추가한다.

```
kcc@linux:~

# vi /etc/rc.d/init.d/syslog
if [ -f /etc/sysconfig/syslog ] ; then
        . /etc/sysconfig/syslog
else
        SYSLOGD_OPTIONS="-m 0"
        KLOGD_OPTINS="-2"
fi
RETVAL=0
umask 077
start() {
        echo -n $"Starting system logger: "
        daemon syslogd $SYSLOGD_OPTIONS -r -h
        RETVAL=$?
        Echo
```

- m 0 : 지정한 분동안에 《MARK》라고 리력파일에 기록한다. 0이면 기록하지 않는다.
- r : 인터네트도메인소케트를 리용하여 망에서 통보문을 받는 설정값이다.
- h : 기본적으로 syslogd는 원격호스트에서 받은 통보문을 리력기록으로 전송하지 않는다. 이 설정값을 사용하여 받은측의 리력파일에 기록한다.

수정되였다면 데몬을 다시 시작해본다.

```
kcc@linux:~

# /etc/rc.d/init.d/syslog restart
핵심부관련기록을 완료함:              [ 확인 ]
체계기록을 완료하고있습니다:          [ 확인 ]
체계기록을 시작하고있습니다:          [ 확인 ]
핵심부관련기록을 시작함:              [ 확인 ]
```

xinetd

인터네트고급데몬

고급데몬

대부분의 망봉사기들은 요청을 기다리며 대기하고있는 보조프로세스(subprocess)들을 가지고있지 않다. 이것은 inetd라는 인터네트고급봉사기가 대신하게 되는데 inetd는 설정된 모든 망포구들을 대기하고있다가 요청이 오면 해당하는 봉사기를 기동한다. RedHat 7.0이전판본에서 inetd의 설정파일은 /etc/inetd.conf이다. RedHat 7.0이상의 판본에서는 /etc/xinetd.d파일등록부에 개개의 파일로써 설정가능하다.

```
kcc@linux:~
# ls /etc/xinetd.d/
amanda        daytime       finger  klogin        rexec   talk
amandaidx     daytime-udp   gssftp  krb5-telnet   rlogin  telnet
amidxtape     dbskkd-cdb    imap    kshell        rsh tftp
chargen       echo          imaps   linuxconf-web rsync time
chargen-udp   echo-udp      ipop2   ntalk         sgi_fam time-udp
camsat        eklogin       ipop3   pop3s         swat
```

매 xinetd에서 관리하는 봉사들을 볼수 있다.

실례로 telnet봉사를 한번 열어보도록 하자.

```
kcc@linux:~
# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#     unencrypted username/password pairs fo authentication.
service telnet
{
        disable         = no
        flags           = REUSE
        socket_type     = stream
        wait            = no
```

```
        user            = root
        server          = /usr/sbin/in.telnetd
         log_on_failure +=USERID
}
```

disable설정값이 no로 설정되어있는가를 확인한 다음에 설정을 no로 수정한다.

설정이 끝나면 /etc/rc.d/init.d/xinetd restart로 데몬을 다시 시작하여 telnet봉사를 한다.

제 3 장
쉘프로그람작성
(SHELL PROGRAMMING)

어떤 명령을 반복적으로 내리거나 여러 명령어를 동시에 사용해야 하는 경우에 많은 시간이 걸리는것은 당연한 일이다. 그러나 실력있는 사람들은 쉘스크립트를 만들어 한번에 실행시킬수 있다. 여기에서는 쉘프로그람작성의 전반에 대한 설명은 략하고 일반적으로 사용할수 있는 스크립트를 작성하는데서 필요한것에 대하여 설명한다.

쉘프로그람작성 bash(Bourne Again SHell)

쉘(Shell)이란 ?

조작체계(OS)와 사용자를 련결해주는 다리의 역할을 한다. Windows에서 서류철의 목록을 확인하려고 할 때에 파일관리자를 열어보려는 서류철을 선택하면 조작체계는 그것을 처리하여 보여주게 되는데 쉘도 마찬가지이다.

Linux 및 UNIX의 쉘은 본문환경에서 사용자가 원하는 작업을 명령하고 그 명령을 조작체계를 통해 수행하여 다시 사용자에게 결과를 보여준다.

bash는 그중 가장 많이 사용하는 쉘중의 하나이다.

여기에서는 쉘을 처음으로 익히고 쉘스크립트를 사용하려는 사람들을 위해 먼저 쉘스크립트의 학습목적에 대해 언급하기로 한다.

배우기 쉬운 쉘스크립트

처음으로 쉘에 대하는 사람도 간단한 쉘스크립트의 작성은 능히 할수 있다. 물론 효과적인 기능의 스크립트를 작성하기 위해서는 많은 경험과 지식이 필요하지만 많은 경우에 사용자가 필요한 몇개의 명령문을 보여주는것만으로도 스크립트를 만들어 사용할수 있다. 단순반복적인 작업을 하는 몇개 명령문을 등록하고 사용해보자. 현재 등록부의 권한을 750으로, 사용자를 pirania로, 집단을 fish로 변경해보자.

```
# chmod 750*
# chown pirania:fish *
```

이 작업을 자주 사용해야 한다면 편집기를 열어 다음과 같이 작성해줄수 있다. 이것은 단 하나의 명령으로 끝낼수 있다.

```
# !/bin/bash
chmod 750*
chown pirania:fish*
```

쉘스크립트는 어려울것이 없다. 단순한것부터 시작하여 재미를 느끼게 되면 곧 스크립트를 본격적으로 활용할수 있다.

사용하기 편리한 쉘스크립트

쉘스크립트는 매우 다양한 용도에 편리하게 사용할수 있다. 우의 실례에서 본바와 같이 여러 명령문을 한번에 실행시킬수 있고 자료의 추출과 쉽게 체계를 관리하는데도 사용하는 등 작업을 효률적으로 할수 있게 한다.

또한 기능을 쉽게 추가할수 있으며 오유수정을 편리하게 할수 있다. 우의 례제를 실행한 후에 현재 등록부의 파일들이 제대로 변경되였는가를 확인해보자. 이때에는 ls -al명령을 사용해야 하는데 이 명령을 우의 스크립트에 포함시키면 한번의 실행에 파일목록까지 확인하는 기능을 추가한 스크립트가 완성된다.

```
# !/bin/bash
chmod 750*
chown pirania:fish*
ls -al
```

재미있는 쉘스크립트

GUI환경만 사용하다가 처음으로 쉘을 대하는 사용자들의 경우 처음에는 눈에 설고 딱딱하게 느껴질수도 있다. 처음에는 앞에서 설명한것과 같은 단순한 스크립트부터 작성해보고 반복실행해보면서 점차 숙련시키면 곧 재미를 느낄수 있다. 스크립트는 한번의 실행만으로 전체작업을 수행할수 있게 한다. 또한 스크립트를 만드는데 특별한 지식이 필요한것도 아니며 결과를 즉시에 볼수 있으므로 매우 흥미가 있다.

이와 같이 쉘스크립트는 Linux 및 UNIX체계를 쉽고 편리하게 관리할수 있도록 도와준다. 쉬운것부터 시작해보는것이 좋으며 이후에 간단한 명령문들과 루틴이 모여 더 높은 기능을 수행하는 스크립트를 완성할수 있다.

화면에 《hello world !》를 출력하기

많은 책들에서 개발언어에 의한 프로그람작성에서 처음으로 보여주는 실례가 《hello world》를 출력하는것이다.

여기에서도 화면에 《hello world!》를 출력하는 간단한 스크립트를 만들어보자.

```
kcc@linux:~
# !/bin/bash
echo hello world
```

이 스크립트는 두 행으로만 되여있지만 스크립트로서 갖추어야 할 기본적인 요소를 갖추고있다. 먼저 첫번째 행에서는 이 파일을 실행하기 위해 필요한 프로그람의 경로를 나타내였다. 잘 아는것이지만 bash란 쉘의 일종이며 /bin/bash는 이 쉘의 실행파일이 있는 경로이다. 쉘은 명령문해석기로 들어오는 명령을 해석하여 이것이 내부명령문이면 즉시에 실행하며 외부명령문이면 해당하는 실행파일을 찾아 주기억기에 필요한 부분을 적재한다. 이 행이 없으면 다음행의 명령을 실행할수 없다.

두번째 행은 실제적인 명령을 내리는 부분이다. 여기에서는 《Hello World!》라는 문자렬을 말단에 출력하라는 명령을 주었다.

파일서술자(Descriptor)와 방향재설정(Redirection)

파일서술자는 보통 Stdin(표준입력), stdout(표준출력), stderr(표준오유) 등의 세가지로 나누어 볼수 있다. 여기서 표준출력이란 프로그람의 실행결과로서 화면에 나타나는 요소들을 의미하며 표준입력이란 어떤 프로그람을 실행하기 위해 꼭 필요한 요소를 프로그람을 실행할 때에 함께 입력해주는 것이다. 이때 >이나 < 와 같은 기호를 사용하여 이와 같은 파일서술자를 다른 파일서술자로 바꾸어줄수 있다.

먼저 다음과 같은 경우를 생각해 보자.

```
# ls -l > ls-l.txt
```

우의 명령은 ls -l을 실행한 결과를 파일에 보관하라는것이다. 이 내용을 실행하면 화면에 ls -l명령의 실행결과가 출력되는 대신 ls-l.txt 라는 이름의 파일이 생긴다. 이 파일을 열어보면 원래 ls -l을 실행했을 때의 결과가 그대로 보관되여있다. 이런 이름의 파일이 원래 있었다면 파일의 앞부분에 그대로 출력내용이 덧쓰기되므로 주의해야 한다.

표준출력이라면 우와 같은 방법만으로도 충분하지만 표준오유의 경우에는 다음과 같이 보관할수 있다.

```
# grep da* 2> grep-errors.txt
```

앞에서 표준출력을 파일로 보냈을 때와 마찬가지로 grep-errors.txt파일이 만들어지며 화면출력대신 이 파일안에 오유통보문이 기록된다. 여기에서 수자 2는 표준오유를 의미하는 파일서술자이다. 2가 들어가지 않으면 표준출력이 보관된다.

파이프(pipes)

파이프는 매우 간단한 방법으로 한 프로그람에서의 출력을 다른 프로그람의 입력으로 보낸다. 이렇게 흘러보내기때문에 파이프(pipe:도관)라는 이름으로 불리운다. 그러면 다음의 실례를 보자. 이 방법은 ls -l*.txt를 사용하는것보다는 까다롭지만 파이프의 사용에 대해서는 충분히 보여줄수 있다.

```
# ls-l | grep "\.txt$"
```

여기서 ls -l의 결과는 grep으로 넘어가 《\.txt$》라는 조건에 맞는 값만을 화면에 출력한다. 이 결과는 ls -l *.txt와 같다.

변수

다른 프로그람작성언어를 사용할 때와 마찬가지로 쉘프로그람작성을 할 때에도 변수를 사용할수 있다. 또한 자료형식을 미리 정할 필요없이 수자나 문자, 또는 문자렬을 지정할수 있다. 어떤 변수가 처음 사용될 때에 참조가 생

성되므로 변수를 따로 선언할 필요는 없다. 그러면 문자렬변수를 선언하여 이 변수에 담긴 문자렬을 화면에 출력해 보자.

```
#!/bin/bash
STR="Hello World!"
echo $STR
```

2번째 행에서 STR이라는 이름의 변수가 생성되여 《Hello World!》를 받았다. 이 변수에 들어있는 값을 사용하기 위해서는 $를 변수이름앞에 사용하여 이것이 변수라는것을 알려주어야 한다. 변수이름앞에 $를 빼놓은 경우에는 예상했던것과는 다른 결과가 나온다. 이런 경우에 쉘은 이 변수자체를 문자렬로 인식해버리므로 주의해야 한다.

조건문

조건문은 어떤 작업을 수행하는가, 또는 하지 않는가의 문제를 표현하기 위한 방법이다.

조건문은 다양한 형식으로 표현할수 있지만 가장 널리 사용하는것은 다음과 같은 형식이다.

```
if    조건
      then  실행
```

《실행》은 《조건》이 만족되였을 때에만 진행된다. 이것의 발전된 형태는 다음과 같다.

```
if    조건
       then  실행1
else  실행 2
```

이것 역시 다른 프로그람작성언어에서 사용하는것과 같은 방식이다.

《조건》이 만족되면 《실행1》이 실행되고 그렇지 않으면 《실행2》가 실행되는 방식이다. 우의것보다 더 개선된 형태를 아래에 주었는데 이것은 여러개의 조건을 사용할수 있는 방법이다.

```
if     조건1
        then  실행1
else  if  조건2
        then  실행2
else  실행3
```

그러면 간단한 문법형식을 보도록 하자.

if를 사용할 때에는 다음과 같이 작성한다.

```
if [조건]
then 조건이 《참》일 때에 실행할 코드
fi
```

순환문

순환문을 사용할수 있는 방법은 for, while, until 등과 같이 다양하지만 여기에서는 for를 사용하는 방법만 보기로 한다.

for를 사용한 순환문의 경우, 일반적인 프로그람작성언어를 사용할 때와 약간 다른점이 있다. 례컨대 문자렬에서의 매 단어를 지나쳐가며 문자렬이 끝날 때까지 순환을 사용할수도 있다.

```
#!/bin/bash
 for i in $(ls); do
        echo item : $i
done
```

두번째 행에서 특수한 변수를 하나 볼수 있는데 이것은 ls의 실행결과를 통채로 변수로 받고있다. 여기에서 $i는 ls의 실행결과를 순서대로 한 단어씩 받는다. 세번째 행은 순환(loop)이 진행되는동안 실행할 내용이다.

여기서는 한행짜리로 되어있지만 필요에 따라 몇행이라도 추가할수 있다. 필요한만큼 추가해준 다음에 마지막행에 done을 입력하여 순환이 끝이라는 것을 알려주면 된다. 마지막 행의 done은 $i가 지금 받아서 사용한 변수는 페기하고 새로운 변수를 받아야 한다는 의미이다. done이전에 있던 내용은 실행되였고 이제 다시 새로운 변수를 넣어 같은 내용을 진행하게 된다.

이 스크립트는 매우 단순한것이만 순환에 필요한 내용은 다 갖추고있다. 이 내용만 제대로 리해하여도 기본적으로 순환을 사용할수 있으리라고 본다.

또한 일반적인 프로그람작성언어에서 사용하는 for에 대해서도 한번 보기로 하자.

```
#!/bin/bash
for i in 'seq 1 10';
do
        echo $i
done
```

쉘스크립트오유수정

어떤 프로그람이 리해할수 없는 동작을 하는 경우, 프로그람의 첫 행을 다음과 같이 수정한다.

```
#!/bin/bash -x
```

이런 설정값은 실행과정에서의 정보를 하나하나 출력해주어 오유수정에 도움을 준다. 매 행이 실행될 때의 반응을 모두 볼수 있으므로 어느 부분에서 문제가 발생하였는가를 확실하게 확인할수 있다.

# #

주해

주해나 무시할 명령문행을 표시한다.

형식 : # 명령문행 또는 주해

행앞에 사용되여 그 행의 명령을 무시한다. 중간에 주해로 스크립트에 설명을 첨부할 때에 사용하거나 림시로 무시할 명령문행에 적용한다.

아래의것은 어떤 스크립트파일의 일부분이다. 이 파일의 매 부분이 어떤 역할을 하는가를 보자.

```
#!/bin/bash
# 그림수집 스크립트
filename ="http://yourURL/yourPWD/FILENAME.jpg"
# 수집해올 그림의 경로를 적는다.
 wget $filename
# 지정한 경로에서 파일을 복사해온다.
```

이 스크립트에서 매 행을 설명하는 부분은 실지 스크립트가 구동될 때에는 무시하고 넘어가야 한다. 매 설명부분의 앞에는 #표시가 들어있어 이 표시가 있는 경우에는 그 행의 내용을 실행하지 않고 뛰여 넘는다.

#!

쉘선언

어떤 쉘로 동작할것인가를 지정한다.

형식 : #![쉘의 절대경로]

스크립트의 시작부분에 사용되여 스크립트가 어떤 쉘에서 동작할것인가를 지정한다. 물론 UNIX, Linux의 쉘프로그람작성은 어느 쉘을 사용하여도 기본적으로는 비슷하지만 세부적인 부분에서는 달라지는 점이 있으므로 정확하게 쉘을 지정해주어야 한다. /bin/sh로 지정하는 경우에는 대부분 원래의 쉘이 아닌 체계의 기본쉘에 대한 련결(link)로 지정되여있으므로 기본쉘을 사용할수 있다.

#!/bin/bash	bash쉘로 동작
#!/bin/tcsh	tc쉘로 동작
#!/bin/sh	대다수의 경우에 체계기본쉘로 동작
#!bin/csh	c쉘로 동작

break

정지

순환을 빠져나간다.

형식 : break[n]

스크립트 for, while, until 등의 순환(loop)안에 사용되여 순환을 빠져나간다. 아래의 스크립트를 보자. 아래에 준것은 while문으로 이루어진 순환을 보여주는 례제이다. break를 사용하지 않는 경우 변수 j가 1부터 3까지 출력되게 되며 break를 사용하는 경우 1을 출력하고 다시 while에로 돌아가 변수를 조건에 비교하지 않고 순환을 빠져나가게 된다.

break 스크립트(파일명 : breakscript)

스크립트	설　　명
#!/bin/bash	bash쉘로 스크립트가 실행
j="0"	변수 j를 0으로 선언
while ["$j"!="3"]	while순환으로 변수 j가 3이 아니면<!=> 《참》이다.
do let "j +=1" echo "$j" # break 1	while이 《참》인 경우에 do를 실행한다. 만일 《break》가 없는 경우를 검사하기 위해 앞에 《#》로 주해처리한 후에 실행한다. 또한 《#》를 삭제하여 break를 넣고 검사한다.
done	변수 j가 3인 경우, done아래를 실행한다.
echo	《스크립트완료》문자렬을 출력한다.

스크립트의 실행

```
#./breakscript
```

우의 명령의 실행결과는 break의 사용여부에 따라 다음과 같이 나타난다.

break를 사용하지 않은 경우

```
1
2
3
스크립트완료
```

break를 사용한 경우

```
1
스크립트완료
```

case

다중분기문

case조건을 검사한다.

형식 :

```
case 문자렬
in
비교문자렬
        command
esac
```

case는 문자렬을 비교하여 문자렬과 같은 문자렬아래의 명령문을 실행한다. case는 문자렬을 비교하여 명령을 실행한다.

case스크립트(파일명 : casescript)

스크립트	설 명
#!/bin/bash	bash쉘로 스크립트를 실행
case	"$1" in$1을 문자렬로 받아 비교한다. $1은 이 스크립트를 실행할 때에 받아오는 첫번째 인수를 의미한다. 례들어 #>./casescript case1로 실행하면 case1을 $1로 받아오게 된다.
case1) echo "case1 - 가는길 험난해도 웃으며 가자!" ;; case2) echo "case2 - 영웅이 많은 영웅의 나라" ;; *) echo "안녕하십니까?" ;;	$1이 받아온 문자렬과 비교하여 같은 문자렬이 있는곳에 명령문을 실행한다. *)은 모든 다른 문자렬을 의미한다. 같은 문자렬이 없는 경우, *)곳에 있는 명령을 실행하게 된다.
case	case를 완료한다.

스크립트 실행

다음과 같은 형태로 실행한다.

```
# ./casescript 문자열
```

직접 실행해보자. 결과는 다음과 같이 나온다.

```
kcc@linux:~
# ./casescript case1
case1 - 가는길 험난해도 웃으며 가자!
# ./casescript case2
case2 - 영웅이 많은 영웅의 나라
# ./casescript case 3
안녕하십니까?
# ./casescript
안녕하십니까?
```

dirs

기억등록부의 보기

현재 기억하고있는 등록부를 보여준다. pushd/popd연산에 관련있다

형식 : dirs[설정값]

+entry : 등록부목록의 시작부터 entry번째의 목록을 보여준다(시작은 0).
−entry : 등록부목록의 끝부터 entry번째의 목록을 보여준다.
-1 : 긴 목록을 보여준다.

pushd를 리용하여 등록부를 기억시킨다. popd로 기억된 등록부로 이동하는데 앞서 이동했던 등록부목록을 보려고 할 때에 사용한다. 설정값을 리용하여 시작으로부터 몇번째라고 지정해주면 그 위치에 기억된 등록부가 출력된다.

push를 리용한 등록부보관

```
# pushd /tmp
# push /home/first
# push /root
```

마지막부터 두번째 기억된 등록부를 보여주기

마지막부터(《-》 설정값사용)두번째(1) : -1

```
kcc@linux:~
# dirs -1

/home/first
```

exit

완료

쉘스크립트를 완료한다.

형식 : exit[n]

스크립트를 완료한다. exit 0은 성공적인 완료를 나타내며 0이 아닐 때에는 실패를 나타낸다. n값이 없는 명령으로 쓰이면 log out와 같은 기능을 하여 말단을 완료시키거나 체계를 탈퇴하도록 한다.

exit스크립트(파일명 : exitscript)

스크립트	설　명
#!/bin/bash	bash쉘로 스크립트가 실행
echo 《스크립트시작》	《스크립트시작》을 화면에 출력한다.
#exit 0	현재 #표시로 주해표시가 되여있다. #을 제거하면 스크립트를 완료하고 나가게 된다.
echo 《스크립트완료》	《스크립트완료》를 화면에 출력한다.

스크립트실행

먼저 exit를 사용하지 않은 경우를 생각해보자.

```
kcc@linux:~
# ./exitscript
스크립트시작
스크립트완료
```

다음으로 exit를 사용한 경우를 생각해 보자.

```
kcc@linux:~
# ./exitscript
스크립트 시작
스크립트 완료
```

export

변수지정

변수를 설정한다.

형식 : export[설정값]변수명=설정

-f : 함수를 가리킨다.
-p : export목록을 출력한다.
-n : export 속성을 제거한다.

export는 변수를 지정하는 명령문이다.
현재 체계에 설정되어있는 변수목록을 볼수 있다.

```
kcc@linux:~
# export
declare -x BASH_ ENV="/home/office/officemaster/.bashrc"
declare -x HISTSIZE="1000"
declare -x HOME="/home/office/officemaster"
declare -x HOSTNAME="office.kcccom.com"
declare -x HOSTTYPE="i386"
```

체계의 qt등록부를 가리키는 변수인 QTDIR를 설정해보자.

```
kcc@linux:~
# export QTDIR="/usr/lib/qt-3.0.4"
# export | grep QTDIR
declare -x QTDIR="/usr/lib/qt-3.0.4"
```

export의 목록을 파이프(|)로 받아 grep명령문으로 QTDIR문자렬을 검사하여 해당한 행만을 출력한다. export되여있음을 확인할수 있다.

for

for순환

for를 리용한 순환문이다.

형식 :

```
for x in 목록
    do
        지령행
done
```

for는 목록에 있는것을 하나씩 x에 넣어 do를 실행한다. 목록을 모두 do에 명령행으로 실행하고 빠져나간다.

스크립트	설　　명
#!/bin/bash	bash쉘로 스크립트가 실행
for i in test1 test2 test3	for문으로 test1부터 test3까지 차례로 i변수에 넣으면서 순환하게 된다.
do echo "$i"	i에 받은 값을 do아래의 명령에서 실행한다. echo로 i가 무엇인가를 출력한다.
done	for순환을 마치고 done아래를 실행한다.

스크립트실행

우의 스크립트를 실행해보자.

```
kcc@linux:~

# ./forscript
test1
test2
test3
```

순환에 들어가는 목록은 우와 같이 지정해줄수 있으며 sed나 awk 등의 행편집기에서 한행씩 입력받아 사용한다.

다음의 실례는 ftp접근리력을 가지고있는 /var/log/xferlog파일을 검색하여 first사용자가 접근한 파일을 따로 복사하는 스크립트이다.

awk에 의해 추출(filtering)되여 목록안의 한행씩 j가 받아 do에서 사용되게 된다.

```
kcc@linux:~

#!/bin/bash

mkdir File ⇐ 파일등록부를 만든다.
for j in awk '/first/ {print $9}' /var/log/xferlog
    do
cp —parents $j File/ ⇐ $j내용을 파일등록부에 경로와 함께 복사한다.

    done
```

help

내부명령문도움말

내부명령문의 도움말을 보여준다.

형식 : help[설정값]문자열

-s : 지정한 문자열이 포함된 명령문을 모두 보여준다.

help는 내부명령문의 도움말을 보여준다.

```
kcc@linux:~

# help fg
fg: fg [job_spec]
    Place JOB_SPEC in the foreground, and make it the current job. If
    JOB_SPEC is not present, the shell's notion of the current jo
    b is used.
```

지정된 문자열이 포함된 내부명령문을 검색하기 위해서는 -s설정값을 사용한다. 검색된 내부명령문과 간단한 사용법을 보여준다.

```
kcc@linux:~

# help -s time
time: time [-p] PIPELINE
times: times
```

history

이전 명령문

이미 사용한 명령문(history)을 보여준다.

형식 : history[수자][설정값]

수자 : 최근에 사용한 명령문을 수자만큼 출력한다.
-c : 모든 history내용을 삭제한다.
-d offset : history목록안의 삭제하려는 번호를 지정하여 삭제한다.
-w 파일명 : 현재의 history내용을 지정한 파일에 보존한다.
-r 파일명 : history파일을 읽어 보여준다.
-a 파일명 : history파일에 이 section을 추가한다.
-n 파일명 : history 파일에서 읽지 않은 목록을 읽는다.

history는 사용한 명령문을 보여준다. history는 HISTFILE에 보관된다. HISTFILE의 크기는 HISTFILESIZE를 넘지 않는다. 그러면 현재 체계의 지정된 HISTFILE과 HISTFILESIZE를 확인해보자.

```
kcc@linux:~
# echo $HISTFILE
/root/.bash_history
# echo $HISTFILESIZE
1000
```

history파일로는 /root/.bash_history가 지정되여있다. 또한 history파일크기로는 1000이 설정되여있는데 1000은 history에 쌓여있는 명령문개수이다. 아래와 같이 최근에 사용한 명령문 5개를 지정하여 볼수 있다.

```
kcc@linux:~
# history 5
1048  ls -a
```

```
1049 history
1050  ssh -l root 168.126.198.50
1051  !
1052  history 5
```

사건지시자(Event Designators)

사건지시자는 리력을 검색하여 사용하였던 명령을 실행한다.

지 시 자	설 명
!!	바로 전에 실행한 명령문을 실행한다. !-1과 같다.
!n	N번 명령행의 명령을 실행한다. history에서 나오는 명령문목록의 번호를 지정한다.
!-n	최근 실행한 명령으로부터 그 이전으로 순서를 부여하여 n번째 명령문을 실행한다.
!문자렬	문자렬로 시작하는 가장 최근의 명령을 실행한다. 가장 최근에 사용된 ssh명령을 실행한다. #!ssh ssh -lroot www.linuxroot.co.kp root@www.linuxroot.co.kp's password:
!?문자렬[?]	문자렬을 포함하는 가장 최근의 명령을 가리킨다. 가장 최근에 사용된 www.linuxroot.co.kp포함된 명령문을 실행한다. #!?www.linuxroot.co.kp ssh l root www.linxroot.co.kp root@www.linxroot.co.kp' s password:
^문자렬1^ 문자렬2^	바로 전의 명령문에 포함된 《문자렬1》을 《문자렬2》로 치환하여 명령한다. 《!!:s/문자렬1/문자렬2/》와 같다. 바로 전의 명령문이 《ls-al》이였다. 《-al》설정값을 《-a》설정값으로 변경하여 실행해보자. # ^-al^-a^ ls -a
!#	지금까지 입력한 전체 명령행.

if

if조건문

if 조건문으로 명령문을 실행한다.

형식 :

```
if test 조건1
      then
            지령행 1
elif test 조건2
            지령행2

...
else
            지령행
fi
```

if조건문은 if의 조건을 검사하여 《참》인 경우에 then이하를 실행하며 《거짓》인 경우에 else이하를 실행한다. 조건검사부분은 본문의 《test》를 참고한다.

if스크립트(파일명 : ifscript)

스크립트	설　　명
#!/bin/bash	bash쉘로 스크립트가 실행
If [-f"/root/.bash_history"]	if조건문으로 조건을 검사한다. -f는 test명령문으로 파일이 있으면 《참》을, 파일이 없으면 《거짓》을 반환한다.
then echo"history파일이 존재합니다. "	history파일이 있는 경우에 《history파일이 존재합니다.》를 출력한다.
fi	조건문을 마친다.

스크립트실행

```
kcc@linux:~
#>./ifscript
리력파일이 존재합니다.
```

let

산수연산

수학연산자를 리용하여 산수연산을 한다.

형식 : let 인수[인수...]

let는 수학연산자를 리용하여 산수연산을 한다. 스크립트내에 let가 없이 산수연산을 사용하면 명령행으로 인식하여 요구하는 결과를 얻을수 없다.

순환문에 자주 사용되는 가장 간단한 변수연산을 아래에 보여주었다.

```
kcc@linux:~
#!/bin/bash
i=" "
let "i=i+10"
echo "$i"
```

스크립트의 실행

```
kcc@linux:~
#. /letscript1
10
```

산수연산의 우선순위

표에서 아래로 갈수록 우선순위가 낮아진다.

산수연산	설　　명
+, -	부호(plus, minus)
!, ~	론리적, 비트부정
*, /, %	곱하기, 나누기, 나머지연산
+, -	더하기, 덜기
《, 》	왼쪽비트반전, 오른쪽비트반전

<table>
<tr><td><=, >=, <, ></td><td>비교연산자</td></tr>
<tr><td>=, ! =</td><td>같다, 같지 않다</td></tr>
<tr><td>&</td><td>비트 AND</td></tr>
<tr><td>^</td><td>비트 XOR</td></tr>
<tr><td>|</td><td>비트 OR</td></tr>
<tr><td>&&</td><td>론리적</td></tr>
<tr><td>||</td><td>론리합</td></tr>
<tr><td>=, *=, /=, %=, +=, -=, «=, »=, &=, ^=, |=</td><td>지정</td></tr>
</table>

while

while순환문

while을 리용한 순환문이다.

형식 :

```
while
        do
                지령행
        done
```

while을 리용한 순환문으로서 조건이 《참》이면 do를 실행하여 《거짓》이 나올 때까지 순환하게 된다. test명령문을 리용한 조건검사도 실행한다(test 명령문을 참고할것).

while스크립트(파일명: whilescript)

스크립트	설　　명
#!/bin/bash	bash쉘로 스크립트가 실행
j="0"	변수 j를 0으로 선언
while["$j" !="3"]	while순환으로 변수 j가 3이 아니면(!=) 《참》이다.
do 　　let "j +=1" 　　echo "$j"	while이 《참》인 경우에 실행한다. let명령은 수식을 처리한다. echo로 현재 변수 j의 값을 보여준다. done변수 j가 3인 경우에 done아래를 실행한다.
echo "스크립트완료"	《스크립트완료》문자렬을 출력한다.

스크립트실행

```
# ./whilescript
```

제 4 장
VI 편집기(VI EDITOR)

어떤 환경의 UNIX에서도 사용되고있는 vi편집기는 처음에 낯설고 어려워보이지만 익숙해질수록 빠르고 능률적으로 작업할수 있는것으로 하여 우수한것으로 평가되고있다. 여기에서는 원본의 vi와 다양한 vi들에서 사용하고있는 많은 단축키들과 vi를 사용할 때에 알아두어야 할 정보들에 대하여 소개설명하였다.

VI편집기의 사용해설과 명령들

vi란?

vi는 Unix와 Linux체계에서 가장 널리 쓰이는 본문편집기이다. 많이 쓰이는 리유는 기본적인 말단에서 동작하는 빠른 편집기라는 점도 있지만 무엇보다도 본문편집기로서 갖추어야 할 최상의 기능을 가지고있기때문이다. 그러한 기능들의 구현이 명령문입력형식으로 되여있기때문에 GUI편집기에 익숙된 사용자들의 경우에 처음에는 사용불편함을 느낄수 있지만 말단본문환경에서 효과적인 작업을 하는데는 최상의것이라고 할수 있다.

vi편집기(vi Editor)에 대한 설명에서는 대부분의 사용자들이 Windows환경의 편집기에 더 익숙되여있기때문에 GUI환경의 편집기를 비교하면서 설명해나가는 방법으로 한다. 설명의 마지막부분에 단락내용을 정리하여 기능을 잊어버린 경우에 빨리 찾아 사용할수 있도록 하였다.

또한 많은 기능의 라렬식설명보다는 많이 사용되는 몇개의 기능에 대해 요약설명을 기본으로 하였다. vi편집기는 입력과 보관기능만 알아도 사용할수 있으며 편리한 많은 기능들은 점차로 익혀가면 쉽게 말단편집기를 사용할수 있다.

vi 실행

vi편집기는 말단에서 실행명령을 통해 사용한다.

vi의 실행명령은 새 문서열기, 편집문서열기, 문서복구로 구분할수 있다.

새 문서열기

```
# vi
# vi 새 파일이름
```

새로운 문서에 입력하려고 할 때에는 《vi》를 실행하거나 《vi 새 파일명》으로 새 문서를 시작한다.

《vi》만 실행한 경우 현재 사용중인 vi편집기에 대한 정보를 볼수 있다.

입력/편집방식으로 변환하면 정보는 없어진다.

```
# vi <enter>
```

```
kcc@linux:~

~
~~                      VIM- Vi IMproved
~
~                       version 6.0.93
~                   by Bram Moolenaar et al.
~            Vim is open source and freely distributable
~
~                 Help poor children in Uganda!
~            type :help iccf<Enter> for information
~
~            type :q<Enter>              to exit
~            type :help<Enter> or <F1> for on-line  help
~            type :help version6<Enter> for version info
~
~
~
~
                                 0,0-1          All
```

편집문서열기

vi 파일명 : 편집문서열기
vi+n 파일명 : n번째 행위치로 편집문서를 열기
vi+/실마리어 file : 검색된 실마리어(keyword)위치로 편집문서를 열기
vi -R 파일명 : 읽기전용문서열기

《vi파일명》으로 파일을 연다. 기본적으로 편집이 가능한 형태로 파일을 열며 《vi -R 파일명》인 경우에는 읽기전용으로 파일을 연다.

vi는 파일열기에도 최적화된 작업을 지원하고있는데 파일열기와 페지의 위치결정을 설정값으로 할수 있다. 행수를 지정하거나 실마리어를 검색하여 처음결과가 나온 곳에서 시작할수 있다. 체계를 기동할 때 다중기동할수 있게 해주는 lilo의 설정파일내에서 《hda》라는 단어를 수정하기 위해 문서를 연다.

```
# vi +/hda/etc/lilo.conf
```

```
kcc@linux:~

prompt
timeout=50
default=linux
boot/dev/hda
map=/boot/map
install=/boot/boot.b
message=/boot/message
lba32
image=/boot/vmlinuz-2.4.13-1hl
        label=linux
        initrd=/boot/initrd-2.4.13-1hl.img
        read-only
        root=/dev/hda11
other=/dev/hda1
        label=redhat7.2
other=/dev/hda9
        label=vine2.5
~
~
~
"/etc/lilo.oonf'20L,284C            4,1    All
```

검색된 실마리어는 반전표시로 나오며 반전표시로 나오는 첫 단어가 있는 행에 입력독촉상태(prompt)가 놓인다.

⚠ 경 고

vi편집기를 설명하는 실례로 체계의 /etc아래의 설정파일을 많이 사용하였다. 자주 사용하는 설정파일들을 례를 들어 체계에 보다 친숙해지려는 의미이다. 그러나 이 파일들은 매 부분에서 아주 중요한 역할을 담당하는 파일이므로 중요하게 가동중인 체계에서는 이 파일을 수정할 때에 신중히 하여야 한다. 특별한 목적이 없이 숙련을 위해 여기에서 작업한 경우에는 보관시키지 않고 빠져나오도록 한다.

문서의 보관 및 끝내기

:w(!) : 문서보존(강제지령)
:q(!) : vi끝내기(강제지령)
:wq(!), zz(!) : 보존후에 끝내기(강제지령)

문서작성에 있어서 보관과 완료는 문서의 마무리로서 가장 중요하다. 문서편집중에 [:w]건을 리용하여 문서보관을 하며 [:q]건으로 vi를 끝낼수 있다.

문서를 편집하였으나 보관하지 않고 빠져나오려고 하는 경우 [:q] 키만 사용하면 《No write since last change (use! to override)》라는 수정된 문서를 보관하지 않았다는 통보문이 현시되면서 완료할수 없게 된다. 이런 경우에 :q!와 같이 !를 붙여주면 강제완료할수 있다.

vi -r 파일명 : 문서복구

문서복구

파일편집중에 정전되거나 전원전지가 다 소모된 경우에 편집중인 내용은 swp파일로 보관되여 복구될수 있다.

웨브봉사기설정파일인 /etc/httpd/conf/httpd.conf파일을 수정하던중에 갑자기 전원이 차단된 경우,《vi-r 복구파일명》으로 편집중인 문서를 되살릴수 있다.

#vi -r/etc/httpd/conf/httpd.conf

```
kcc@linux:~

Using swap file "/etc/htpd/conf/.httpd.conf.swp"
Original file "/etc/httpd/conf/httpd.conf
Recovery completed. You should check if everything is OK.
(You might want to write out this file under another name
and run diff with the original file to check for changes)
Delete the .swp file afterwards.

Hit ENTER of type command to continue
```

우와 같이 복구된 파일에 대한 정보가 나온다. 복구를 하기 위한 .swp파일과 원본파일에 대한 정보가 나오며 복구상태에 대한 설명이 나온다.

[Enter]건을 누르면 편집하던 파일이 열린다.

입력 및 편집방식과 명령방식의 변환

vi에는 내용의 입력과 편집을 할수 있는 입력 및 편집방식과 보관하고 탈퇴 등의 기본기능과 여러 편리한 기능을 실행할수 있는 명령방식이 있다.

처음으로 vi를 리용하는 사용자가 가장 어렵게 느껴지는것이 두 방식사이의의 변환이다. 편집기가 실행되였는데 특정한 건을 누르기전에는 입력이 안되고 입력이나 편집한 다음에 보관하고 탈퇴하려면 명령방식으로 빠져나와야하는데 이러한 일련의 변환은 Windows GUI환경의 편집기에 익숙된 사용자인 경우에 어렵게 느껴진다. 사실 Windows GUI환경의 편집기보다 직관적이지 못하고 불편한것은 사실이다. 그러나 입력 및 편집방식과 명령방식사이의 변환에 익숙되면 다른 편집기들보다 편리하다는것을 느끼게 된다.

입력 및 편집 방식

i : 현재 입력독촉상태(prompt)에서 입력방식으로 전환

+입력 및 편집방식은 새로 문서를 작성하거나 기존문서의 내용을 편집할때에 사용한다. 아직 vi편집기에 익숙되지 않은 사용자인 경우에는 《i》를 리용한 입력절환방식만을 사용하여도 충분하다. 많은 입력 및 편집절환방식이 있지만 그것들은 초기입력되는 위치를 좀 더 편집이 쉬운 위치에로 조정해주는 기능에서만 차이날뿐이다. 물론 적합한 입력 및 편집 전환방식을 사용하면 문서편집을 보다 빨리 할수 있다. vi는 명령방식상태로 시작된다. 이 상태에서 《i》를 입력하면 창아래에 《--INSERT—》라고 출력되며 이때 문서의 입력이나 편집을 할수 있다. 다시 명령방식으로 복귀하려면 [ESC]건을 누른다.

봉사기접속IP를 허용하는 /etc/hosts.allow파일을 열어 입력방식으로 전환해보자.

```
# vi /etc/hosts.allow
```

문서가 열리면 명령방식의 상태이므로 아무 건도 입력되지 않는다. 이때 《I》를 누르면 아래에 《--INSERT-》라고 현시되며 이때에 입력이 가능하다.

```
kcc@linux:~

#
# hosts.allow   This file describes the names of tne hosts which
#                are allowed to use the local INET service, as
#                decided by the '/usr/sbin/tcpd' server
#
#

~
~
~
~
~
~
~
~
— INSERT —                    1,1          All
```

a	현재 입력독촉의 한 칸 오른쪽위치에서 입력방식전환
I	현재행의 가장 앞에서 입력방식전환
A	현재행의 가장 뒤에서 입력방식전환
o	현재행의 다음행을 삽입한 다음에 입력방식전환
O	현재행우의 행삽입후에 입력방식전환
s	현재 입력독촉의 한 문자를 삭제한 다음에 입력방식전환
S	현재행을 삭제한 다음에 입력방식전환
R	수정방식(REPLACE)으로 입력방식전환

vi는 입력 및 편집방식으로 변환하는 다양한 건을 가지고있다. 매 건별로 정황에 맞게 입력할수 있는 개개의 기능을 다 가지고있다.

명령방식

명령방식은 GUI환경의 본문편집기에서 안내(Menu)에 등록된 기능들로 보면 된다. 안내띠(Menu Bar)가 없다는 점에서 처음으로 vi를 대하는 사용자의 경우에 어렵게 생각될수 있다. 그러나 안내띠가 있는 GUI환경의 편집

기에서도 조금만 익숙해지면 본문의 복사나 붙여넣기기능을 사용하기 위해 안내띠나 그림기호를 누르지 않고 [Crlr+C]나 [Ctrl+V] 등의 지름건을 리용하는 경우가 많다. vi에서도 이와 마찬가지로 안내띠가 없을뿐이고 같은 기능을 vi에 고유한 지름건을 리용하여 실행하고있다고 생각하면 된다. 또한 이러한 조합은 규칙성이 있으므로 몇가지 건에만 익숙되면 다른 건의 활용을 쉽게 할수 있다.

vi의 시작상태가 명령방식이며 가장 아래부분에 《--INSERT-》, 혹은 《--REPLACE-》가 나오지 않는다면 편집방식이다. 만약 입력 및 편집상태인 경우에 [ESC]를 눌러 명령방식으로 변경할수 있다.

실행취소

u : 실행한 작업을 취소한다.

앞서 실행한 작업을 취소하는 건이다. 삭제하였거나 잘못된 명령이 실행된 경우에 [u]건을 누르면 바로 이전 상태로 되돌린다. 이 건을 명령방식의 처음에 설명한것은 뒤부분에서 나오는 다양한 건으로 여러 검사를 하고 되돌릴수 있도록 하기 위해서이다. 봉사기접속IP를 제한하는 /etc/hosts.deny파일을 연다.

vi /etc/hosts.deny : hosts.deny 파일을 연다.

```
kcc@linux:~

#
# hosts.deny   This file describes the names of the hosts which
#              are *not* allowed to use the local INET services, as
#          decided by the '/usr/sbin/tcpd' server.
#
# the new secure portmap uses hosts.deny and hosts.allow. In
# particular you should know that NFS uses portmap!

~
"/etc/hosts.deny" 9L, 347C             1,1          All
```

[o] 건을 누르면 다음렬이 삽입되면서 입력방식으로 바뀐다. 입력방식에서 《vi editor》라고 입력하고 [ESC] 건을 눌러 명령방식으로 돌아온다.

```
kcc@linux:~

#
 vi editor
# hosts.deny   This file describes the names of the hosts which
#              are *not* allowed to use the local INET services, as
#              decided by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind
# you that the new secure portmap uses hosts. Deny
# and hosts.allow. In particular you should know that
# NFS uses portmap!

~
                                        2,9          All
```

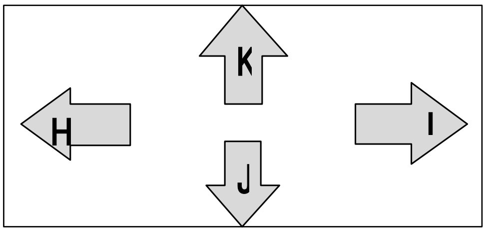
[u] 건을 눌러보자. 그러면 입력방식전으로 돌아온것을 확인할수 있다.

```
kcc@linux:~

#
# hosts.deny   This file describes the names of the hosts
#              which are *not* allowed to use the local INET
#              services, as decided by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind
# you that the new secure portmap uses hosts. Deny
# and hosts.allow. In particular you should know that
# NFS uses portmap!

~
                                        1,1          All
```

이동에 관한 건



k	우로 한행 이동
h	왼쪽으로 한칸 이동
l	오른쪽으로 한칸 이동
j	아래로 한행 이동

이동은 명령방식에서 이루어진다. 물론 방향건으로도 이동이 가능하지만 일부 UNIX체계들의 vi에서 방향건이 동작하지 않는 경우도 있다. 문자로 된 방향건은 수자와 조합하여 여러칸을 이동할수도 있으며 다른 명령문키(c, d)와 합쳐져 강력한 기능을 발휘하므로 그에 익숙되는것이 좋다.

수자와의 조합

[수자][명령문]

수자와 명령건을 붙여쓰는것은 이 명령을 지정된 수자회수만큼 반복하여 실행한다는것을 의미한다. 례를 들어 방향건은 수자와 합쳐져 여러칸이동으로 사용될수 있다.

오른쪽으로 4칸 이동 : 4l
우로 3 행 이동 : 3k

이와 같이 수자와 방향건을 조합하여 사용하면 요구하는 행이나 칸으로 한번에 움직일수 있다. 뿐만아니라 다양한 명령에 대해 이 방법을 사용할수 있다.

0 : 행의 가장 앞으로 이동
$: 행의 맨 뒤로 이동

행안에서의 이동

일반적인 편집기에서 Home, End 등의 기능과 같이 행의 어느 위치에 있든간에 맨 앞과 맨 뒤에로 이동할수 있다.

w : 오른쪽으로 한 단어 이동
b : 왼쪽으로 한 단어 이동

단어단위의 이동

오른쪽으로 3단어 이동 : 3w
왼쪽으로 6 단어 이동 : 6b

단어단위의 이동도 수자와 조합하여 쉽게 사용할수 있다.

G : 해당한 행으로 이동

행이동

66 행으로 이동 : 66G

원하는 행으로 이동할 때에는 행번호와 함께 G를 입력한다.

편집

이동방향건의 조합에 의한 편집

c : 본문삭제후 입력방식절환(Change)
d : 본문삭제
y : 본문복사
p : 복사나 삭제된 내용을 붙여넣기

편집기로서 중요한 기능은 편집기능인데 얼마나 편리하고 신속하게 본문을 편집할수 있는가가 중요하다. 그러한 면에서 vi는 편집하려고 하는 문장뿐만 아니라 단어의 문자 하나까지 편집할수 있다.

삭제와 복사기능의 편집건은 방향건과 같이 사용되여 매 본문의 한 문자에까지 자세한 설정을 할수 있다.

편집건 + 방향건

편집건과 방향건을 조합하면 편집건의 기능에 적용하는 범위를 방향건으로 조절할수 있다.

편집건	방향건	조합건	기능의 설명
c	h(좌), l(우), j(하), k(상)	ch, cl, ci, ck	선택한 방향의 본문을 삭제한 다음 입력방식변환
	W(단어)	cw	선택한 단어를 삭제한 다음 입력방식변환
	c	cc	현재행의 삭제후 입력방식변환
d	h(좌), l(우), j(하), k(상)	dh, dl, dj, dk	선택한 방향의 본문을 삭제
	w(단어)	dw	선택한 단어를 삭제
	d	dd	현재행을 삭제
y	h(좌), l(우), j(하), k(상)	yh, yl, yi, yk	선택한 방향의 본문을 복사
	w(단어)	yw	선택한 단어를 복사
	y	yy	현재행을 복사

proftp봉사기를 설정할수 있는 /etc/proftpd/proftpd.conf를 열어보자.

```
kcc@linux:~

#vi /etc/proftpd/proftpd.conf

ServerName        "Proftpd FTP Server"
ServerType        standalone
DefaultfServer    on
#vi /etc/proftpd/proftpd.conf
```

```
ServerName          "Proftpd FTP Server"
ServerType          standalone
DefaultfServer      on
Port                21
Umask               022
MaxInstances        30
User                nobody
Group               nobody
UseReverseDNS       off

"/etc/proftpd/proftpd.conf" 69L, 1281C  2,15-34   Top
```

첫 행의 ServrName을 변경해보자.

현재 유표는 《proftpd FTP Server》의 P문자에 위치하고있다. 세 단어로 되여있으므로 세 단어를 삭제한 다음에 입력방식으로 변경하면 된다.

삭제한 다음 입력방식변경건(c) + 3 단어(w) : c3w

```
kcc@linux:~

ServeName           ""
ServerType          standalone
DefaultServer       on
Port                21
Umask               022
MaxInstances        30
User                nobody
Group               nobody
UseReverseDNS       off

—INSERT—                  2,15-34      Top
```

삭제한 다음 입력방식으로 변경되여 《first FTP Service》라는 내용을 입력한 후에 [ESC]를 누르면 다시 편집방식으로 돌아온다.

본문 한 문자의 편집

x : 본문 한 문자의 삭제
s : 본문 한 문자를 삭제한 다음 입력방식전환

본문에서 한 문자만 수정할 때에 사용한다.
/etc/proftpd/proftpd.conf파일을 보자.

```
kcc@linux:~

IdentLookups              off
AuthPAMAuthoritative      on
RootLogin                 off
DenyFilter                \*.*/
DeferWelcome              on
TimesGMT                  off
# RateReadBFS             256
# PateReadFreeBytes       5120
# RateReadHardBPS         on

Timeoutidle               0
TimeoutNoTransfer         0
                                    18,1      Top
```

#로 주해가 되여있는것을 볼수 있다. 주해를 제거하려고 할 때에는 #에 유표를 두어 x건으로 한 문자만 지울수 있다. 이 기능도 수자와의 조합으로 1문자부터 적용하려는 수자를 써주는 방식으로 수정할수 있다. 4문자를 삭제하려면 4x라고 입력하여 유표위치로부터 4문자를 지울수 있다.

4 문자의 삭제 (x) : 4x

s도 x와 같이 한 문자를 삭제하지만 s에서는 입력방식으로 전환된다.
기동방식을 결정하는 /etc/inittab파일에서 기동방식을 바꾸어준다.

```
kcc@linux:~

id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

"/etc/inittab" 57L, 1755C,          18,4      20%
```

현재 기동하고 있는 기본방식이 X Window방식인 RunLevel 5로 되여있다. 이것을 말단으로 기동되게 하려면 5를 3으로 변경하면 된다. 현재 유표의 위치에서 s를 누르면 한 문자가 지워지면서 입력방식으로 전환되고 3을 입력한 후 ［ESC］로 빠져나온다.

```
kcc@linux:~

id:3initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait/etc/rc.d/rc 6

"/etc/inittab" 57L, 1755C            18,4       20%
```

참 고

runlevel이란?

기동의 계층을 나누어놓은것이라고 생각하면 된다. 《실행준위》라고 불리운다. 개개의 기동마다 실행되는 프로쎄스를 달리하여 요구하는 작업에 알맞는 기동을 할 수 있다.

0, 1, 6 번 즉 runlevel 0, runlevel 1, runlevel 6 등은 체계에 예약된 실행준위이며 나머지의 실행준위는 아래와 같이 규정되여있는데 프로쎄스의 추가와 삭제가 가능하다.

runlevel 0

체계를 완료할 때에 사용된다.

runlevel 1

이것은 《단일방식기동》(single mode booting)이라고도 불리우는데 이 준위에서는 제일 처음의 체계봉사가 기동되여 뿌리권한을 가진다. 체계의 **관리자암호**(root password)를 잊어버렸을 때에 사용되는 경우가 많으나 중요한 체계에서는 bootloader부분에 암호를 설정해주는 방법으로 막아주어야 한다.

runlevel 2

NFS를 지원하지 않는 다중사용자방식의 기동이다. 망을 지원하지 않을뿐이고 runlevel 3과 거의 같다.

runlevel 3

망을 지원하는 다중사용자방식의 기동이다. X Window를 지원하지 않는 체계에 한한 기본적인 실행준위이다.

runlevel 4

사용자가 정의하는 실행준위이다. 사용자가 요구되는 봉사를 등록하여 사용하면 된다.

runlevel 5

X Window기동을 지원하는 실행준위이다. X Window를 사용중인 사용자의 경우에 이 방식으로 기동된다.

runlevel 6

체계재기동의 실행준위이다.

기타 편집기능

~ : 대소문자의 변환
. : 반복

대소문자를 변경하는 ~도 수자와 조합하여 사용할수 있다.

telnet로 접속할 때에 접속된 봉사기체계의 정보를 볼수 있다. 이 내용이 들어있는 파일인 /etc/issue.net파일을 연다.

```
#vi /etc/issue.net
```

```
kcc@linux:~

welcome to first system
Kernel 2.4.13-1hl on an i686
~
~
~
~
                              1,0-1          All
```

《welcome》을 대문자로 바꾸기 위해서는 《~》가 문자단위로 실행되므로 《7~》에 의해 7문자를 대문자로 변환하게 한다.

```
kcc@linux:~

WELCOME to first system
Kernel 2.4.13-1hl on an i686
~
~
~
~
                              1,0-1          All
```

이 명령을 반복하려면 《.》건을 리용한다. 이것은 방금 실행한 명령을 반복하는 기능을 한다.

```
kcc@linux:~

WELCOME TO FIRst system
Kernel 2.4.13-1hl on an i686
~
~
~
~
                              1,0-1          All
```

《7~》키가 다시 실행되여 다음의 7문자가 대문자로 변경되였다.

검 색

/검색어: 검색어를 찾아 강조체로 표시한다.
n: 다음의 검색을 위해 아래쪽으로 이동
N: 다음의 검색을 위해 웃쪽으로 이동

편집방식상태에서 《/검색어》를 누르면 문서안을 검색하여 찾은 단어를 강조체(highlight)로 표시한다. 검색된 첫번째 단어에서 다음으로 넘어가려면 n을 누르고 반대로 가려면 N을 누른다.

웨브봉사기설정파일인 /etc/httpd/conf/httpd.conf파일에서 《httpd》라는 검색어로 검색하여보자.

```
/httpd [Enter]
```

```
kcc@linux:~
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/etc/httpd"

#
# The LockFile directive sets the path to the lockfile
# used when Apache is compiled with either
# USE_FCNTL_SERIALIZED_ACCEPT or USE_FLOCK
# _SERIALIZED_ACCEPT. Thi directive should normally
# be left  at its default value. The main reason for
# changing it is if the logs directory is NFS mounted,
# since the lockfile MUST BE STORED ON A LOCAL
# DISK. The PID of the main server process is
# automatically appended to the filename.
#
Lockile /var/run/httpd.lock

# PidFile: The file in which the server should
# record its prccess identification number when it starts.
#
PidFile  /var/run/httpd.pid
                                        78,18           3%
```

현재 유표는 가운데의 httpd단어에 있으며 n을 누르면 다음 검색어로 이동하고 《N》을 누르면 그전의 명령문으로 이동하는것을 확인할수 있다.

검색 및 치환

s/검색어/수정어 : 검색한 단어를 수정할 단어로 치환해준다. 현재 행의 첫번째 검색단어만 치환한다.

: %s/검색어/수정어/g : 검색한 단어를 수정할 단어로 치환한다. 모든 행의 모든 검색단어를 치환한다.

: %s/검색어/수정어/gc : 검색한 단어와 수정할 단어를 확인한 다음 치환해준다. 모든 행의 모든 검색단어를 치환한다.

검색한 문자를 다른 문자로 쉽게 바꿀수 있다. vi는 ex에 토대한 편집기이기때문에 ex편집기에서 사용하던 명령문을 거의 사용할수 있다. 특히 원천프로그램을 수정할 때에 변수를 일괄적으로 변경하기 편리한것이 검색 및 치환기능이다.

1. 이동명령문

명 령 문	기 능
문자	
h, j, k, l	왼쪽, 아래, 우, 오른쪽으로 이동
본문	
w, W, b, B,	한단어 오른쪽, 왼쪽으로 이동
e, E	단어의 끝으로 이동
), (	다음의 문장, 이전 문장의 처음으로 이동
}, {	다음의 단락, 이전 단락의 처음으로 이동
]], [[	다음의 절, 이전 절의 시작으로 이동
행	
[enter]	다음 행의 공백이 아닌 처음으로 이동
0, $	현재 행의 처음(0)과 끝($)으로 이동
^	현재 행의 공백이 아닌 처음으로 이동
+, -	다음 행과 이전 행의 공백이 아닌 처음으로 이동
nl	현재 행의 n번째 렬로 이동
H	화면의 맨 웃행으로 이동
M	화면의 중간행으로 이동
L	화면의 맨 아래행으로 이동
nH	화면의 맨 웃행으로부터 n번째 행으로 이동
nL	화면의 맨 아래행으로부터 n번째 행으로 이동
화면이동	
+F, +B	한 화면의 다음으로, 한 화면 이전으로 이동
+D, +U	반 화면의 아래로, 반 화면의 우로 이동
+E, +Y	화면이 한행의 우, 아래로 이동
z	유표가 있는 행을 화면의 맨 첫 행으로 이동
z.	유표가 있는 행을 화면의 중간으로 이동
z-	유표가 있는 행을 화면의 맨 아래로 이동
+L	화면흘리기(scrolling)없이 화면을 재표시
검색	
/pattern	유표의 오른쪽으로 패턴검색
?pattern	유표의 왼쪽으로 패턴검색

n, N /, ? fx Fx tx	마지막으로 검색한 내용을 같은 방향, 혹은 반대방향으로 반복 이전의 검색을 유표의 오른쪽, 왼쪽으로 반복 현재 행에서 문자 x의 오른쪽을 검색 현재 행에서 문자 x의 왼쪽을 검색 현재 행에서 문자 x이전의 문자를 오른쪽으로 검색
Tx ; ,	현재 행에서 문자 x이전의 문자를 왼쪽으로 검색 현재 행에서 이전 검색을 반복 현재 행에서 이전 검색을 반대방향으로 검색
행번호 +G nG G :n	 현재 행수를 표시 n번째 행으로 이동 파일의 마지막행으로 이동 파일에서 n번째행으로 이동
식별위치 mx 'x '' 'x ''	 현재 위치를 x로 표시 유표를 x에로 이동 이전의 식별자나 이전의 문맥으로 이동 x를 포함하고있는 행의 처음으로 이동 이전의 식별자를 포함하고있는 행의 처음으로 이동

2. 편집 명령문

명 령 문	기 능
입력 j, a I, A o, O	 본문을 유표의 앞, 뒤에 입력 본문을 행의 처음, 마지막에 입력 유표가 있는 행의 아래, 우에 새로운 행을 입력
변경 문자변경 cc cmotion C R s S	 cw단어변경 현재 행을 변경 유표와 motion대상사이의 본문을 변경 유표위치부터 그 행의 끝까지 변경 문자를 덧쓰기 문자를 지우고 새로운 본문을 입력 현재 행을 지우고 새로운 본문을 입력
삭제, 이동 x X	 유표가 위치한 문자를 삭제 유표앞의 문자를 삭제

dw dd	단어를 삭제 현재행을 삭제
dmotion D p, P "np	유표와 motion대상사이의 본문을 삭제 유표위치부터 그 행의 끝까지 삭제 유표의 오른쪽, 왼쪽에 지운 본문을 삽입 삭제된 최종 9개의 buffer중 n번째 삭제, buffer의 본문을 유표뒤에 붙인다.
복사 yw yy "ayy ymotion p,P "aP	 단어복사 현재행을 복사 현재행을 a라는 이름의 완충기(buffer이름은 a부터 z까지)에 복사, 대문자의 완충기이름은 완충기에 본문을 추가 유표와 motion대상사이의 본문을 복사 유표의 오른쪽, 왼쪽에 복사한 본문을 삽입 유표의 왼쪽에 buffer a(buffer이름은 a부터 z까지)의 본문을 삽입
기타 명령문 . u, J	 가장 최근의 편집명령을 반복 U 마지막 편집명령을 반환하고 현재행을 복구 두행의 합치기
ex편집명령문 :d :m :co또는 :t :., $d :30, 60m0 :., pattern/co$	 행삭제 행이동 행복사 현재행부터 파일의 마지막까지 삭제 30행부터 60행까지를 파일의 처음으로 이동 유표가 있는 행부터 패턴을 포함한 행까지를 파일의 맨 끝에 복사

3. 완료명령문

명 령 문	기 능
zz	파일을 수정하였을 때에 보관 후 완료
:x	파일을 수정하였을 때에 보관 후 완료
:wq	무조건 파일을 보관 후 완료

4. 보관명령문

명 령 문	기 능
:w	파일보관
:w!	무조건 파일을 보관
:30, 60w newfile	30행부터 60행까지를 newfile로 보관
:30, 60w>>file	30행부터 60행까지를 file에 추가
:w %.new	현재완충기의 파일명을 file.new로 보관
:q	파일완료
:q!	무조건 파일완료
Q	vi를 완료하고 ex로 전환
:e file2	vi를 완료하지 않고 file2를 편집
:r newfile	newfile의 내용을 현재 파일에서 읽기
:n	다음 파일을 편집
:e!	현재 파일을 마지막으로 보관한 상태
:e#	파일을 번갈아 편집
:vi	ex에서 vi를 호출
:	vi에서 ex명령을 호출
%	현재 파일이름(ex명령행에 치환)
#	다음 파일이름(ex명령행에 치환)

5. Solaris vi편집방식의 꼬리표(tag)명령문

명 령 문	기 능
^]	꼬리표파일에서 유표가 있는 곳의 식별자의 위치를 찾아 그 위치로 이동, 만약 tag stacking기능이 동작하고있으면 현재 위치는 자동적으로 tag stack에 보관
^	T꼬리표탄창에서 이전 위치로 전환, 즉 자료 하나를 pop하기

6. 명령문설정값

명 령 문	기 능
vi file	vi에서 파일을 연다.
vi file1 file2	순차적으로 파일을 연다.
view file	읽기전용방식으로 파일을 연다.
vi -R file	읽기전용으로 파일을 연다.
vi -r file	작업중에 중지된 편집상태로 복귀한다.

vi -t tag	꼬리표를 검색하고 검색한 위치에서 편집한다.
vi -w n	창의 크기를 n으로 지정한다. 접속속도가 뜰 때 유용하다.
vi + file	파일을 열 때에 유표가 파일 마지막 행에 위치한다.
vi +n file	파일을 열 때에 유표가 n번째 행에 위치한다.
vi -c command file	파일을 열고 검색명령이나 행번호에 관련된 일반적인 명령을 실행한다(POSIX).
vi +/pattern file	패턴위치에서 파일을 연다.
ex file	파일을 ex편집기로 불러온다.
ex- file < script	정보를 제공하는 통보문과 입력독촉(prompt)은 숨기고 스크립트에서 명령을 받아 파일을 ex편집기로 불러온다.
ex -s file < script	정보를 제공하는 통보문과 입력독촉은 숨기고 스크립트에서 명령을 받아 파일을 ex편집기로 불러온다(POSIX).

7. 기타 ex명령문

명 령 문	기　능
단축명령	
:map x sequence	명령문순차로 건 x를 정의한다. x에는 다중문자로 사용할수 있다
:map! x sequence	삽입방식에서 명령문순차로 키 x를 정의한다.
:unmap x	x의 넘기기(mapping)를 해제한다
:unmap! x	삽입방식에서 x의 넘기기를 해제한다
:ad abbr phrase	addr을 phrase의 간략형으로 정의한다. addr를 삽입방식에서 입력하면 한개의 단어 혹은 문구로 대치된다.
:unab addr	addr의 정의를 해제한다
환경최적화	
:set option	설정값을 활성화한다
:set option=value	설정값에 수값(value)을 할당한다
:set nooption	설정값을 비활성화한다
:set	사용자가 설정한 설정값을 표시한다
:set all	현재의 설정값설정, 기본값, 사용자설정값을 모두 표시한다.
:set option?	설정값을 표시한다
UNIX접근	
:sh	쉘을 호출한다
^D	쉘에서 편집기로 다시 돌아간다
:! command	UNIX명령을 편집기에서 실행한다
:n, m! command	n에서 m행까지의 내용을 UNIX명령으로 추출한다
:r !command	UNIX명령의 출력값을 현재 파일로 읽어들인다

제 5 장
번역기(COMPILER)

Linux에서 무료로 사용할수 있는 개발환경이 아무리 다양하고 강력하여도 제대로 사용하지 못한다면 의의가 없다. Linux를 원만히 실행시키고 활용하자면 번역기(Compiler)에 대해서도 충분한 지식과 실기를 소유하여야 한다.

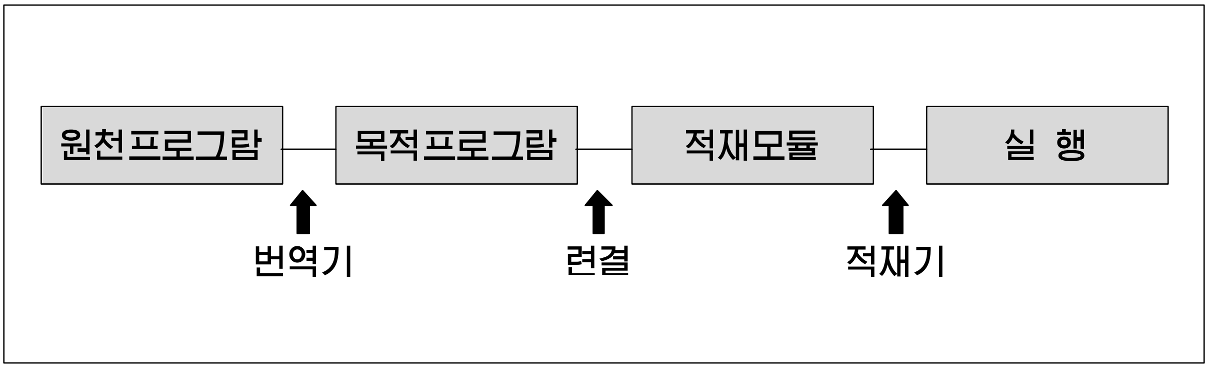
여기에서는 Linux에서 사용할수 있으며 UNIX환경에서도 무료로 구입하여 사용할수 있는 여러 번역기에 대하여 소개설명하였다. 이것은 안정하고 강력한 환경에서 개발자의 자질을 높이는데 도움으로 될수 있다.

번역기(Compiler)

번역기란?

원천프로그람을 목적프로그람으로 번역하는 프로그람이다. Linux에서는 기본적으로 설치를 하면 개발을 위한 여러 번역기가 포함되여있다.

특정한 프로그람언어로 만들어진 프로그람은 사람에게 인식될수 있는 언어로써 콤퓨터는 인식하지 못한다. 따라서 번역(Compile)이라는 과정을 거쳐서 콤퓨터가 인식할수 있는 언어로 다시 바꾸게 되는것이다. 이와 같이 특정한 프로그람을 개발하는 경우에 프로그람개발자가 작성한 원천프로그람을 콤퓨터가 인식할수 있는 기계어, 즉 목적프로그람으로 번역하여 실행가능하게 하는 프로그람을 번역기라고 한다. 한번 번역된 실행프로그람은 프로그람개발자가 작성한 원천프로그람을 수정하여 번역하지 않는 한 계속 사용할수 있다.



1) 번역기(Compiler) : 프로그람개발자가 작성한 원천프로그람을 콤퓨터가 인식할수 있는 기계어인 목적프로그람으로 번역한다.

2) 련결프로그람(Linker) : 기계어로 번역된 목적프로그람을 실행가능한 적재모듈로 번역한다.

3) 적재기(Loader) : 적재모듈을 실행하기 위해 주기억기에 적재한다.

[주의]

목적프로그람을 생성하지 않은 원천코드를 그대로 기계어로 번역하여 실행하는 경우도 있다. 이 작업을 수행하는 프로그람을 해석기(interpreter)라고 한다. 번역과정이 없이 실행할수 있는 스크립트언어 등에서 주로 사용한다.

프로그람작성언어의 발전과정을 보면 크게 1세대의 기계어나 아쎔블러언어, 2세대의 포트란이나 코볼을 거쳐 3세대, 4세대, 5세대의 언어로 구별하여 특징지을수 있다. 번역기는 3세대언어인 C나 파스칼 등을 포함하여 4세대, 5세대의 언어들에도 적용된다.

보통 3세대언어부터는 번역기가 해당 프로그람을 기계어로 번역하는데 이것은 절차적언어로서 고급언어의 단계로 볼수 있다. 4세대언어는 높은 생산성을 배경으로 하는 비절차적언어로서 직결환경에서 응용프로그람을 개발하는 언어의 단계이다. 이러한 언어로서는 자료기지 및 문의언어인 Lotus 1-2-3이나 dBASE 등을 들수 있다. 그리고 5세대언어로는 인공지능분야나 전문가체계에 적용되는 LISP, Prolog 등을 들수 있다.

최근 프로그람작성언어의 흐름을 보면 오브젝트지향프로그람작성언어인 C++, Java, 시각화프로그람작성언어인 Visual Basic, Visual C++ 등의 언어가 널리 쓰이고있으며 최근에는 마이크로쏘프트회사가 내놓은 .NET계렬의 언어가 주목되고있다.

아래에 Linux에서 사용할수 있는 대표적인 번역기인 gcc와 프로그람에 필요한 몇가지 편의프로그람, Linux프로그람에 대한 요약설명을 주었다.

gcc

GCC는 원래 GNU C compiler의 략자이지만 주체88년(1999)년 4월에 《GNU Compiler Collection》의 략자로 바뀌였다. 여기에는 C만이 아니라 C++, 포트란, Object C, Java 등이 포함되여있다.

```
kcc@linux:~
# rpm -qalgrep gcc
gcc-chill-2.96-99
gcc-c++-2.96-99
gcc-objc-2.96-99
gcc-2.96-99
gcc-g77-2.96
gcc-java-2.96
```

GNU C번역기는 리챠드 스톨맨에 의해 만들어졌는데 이것은 품질이 매우 좋으며 이식성이 좋은 C, C++번역기로 평가받고있다. 이 번역기는 먼저 등록기이동언어로 번역한 다음에 그것을 다시 목적하는 구성방식의 기계어코드로 바꿈으로서 다중전단 및 후위처리를 지원하기 위해 설계되고 만들어진것이다.

그러면 현재 설치되여있는 gcc판본에 대해 먼저 확인해보자.

```
kcc@linux:~
# gcc -v
Reading specs from/usr/lib/gcc-lib/i386-redhat-linux/
2.96/specs gcc version 2.96 20000731 (RedhatLinux 2.2  2.96-99hl)
```

gcc의 머리부파일(head file)들이 들어있는 파일등록부는 /usr/include 이다.

그러면 설치되여있는 gcc의 구조를 한번 보자.

```
kcc@linux:~

# rpm -ql gcc
/usr/bin/cc
/usr/bin/gcc
/usr/bin/gcov
/usr/bin/i386-redhat-linux-gcc
/usr/bin/protoize
/usr/bin/unprotoize
/usr/lib/gcc-lib
/usr/lib/gcc-lib/i386-redhat-linux
/usr/lib/gcc-lib/i386-redhat-linux/2.96
```

/usr/bin/cc는 gcc에 기호적으로 련결된 파일이다.

```
kcc@linux:~

# ls -al /usr/bin/cc
lrwxrwxrwx 1 root root 3 12월 14  2001 /usr/bin/cc -> gcc
```

/usr/lib/gcc-lib파일등록부아래에 gcc에 관한 모든 내용이 설치된다.

```
kcc@linux:~

# ls-al /usr/lib/gcc-lib/
합계 80
drwxr-xr-x    4  root   root   4096   11월 17  2001 .
drwxr-xr-x 121  root   root   65536   7월 4  18:06 ..
drwxr-xr-x    3  root   root   4096  12월 14  2001 i386 -glibc21-linux
drwxr-xr-x    4  root   root   4096  11월 17  2001 i386 -redhat-linux
```

/usr/lib/gcc-lib/i386-redbat-linux/2.96파일등록부를 보면 아래와 같은 파일들이 있다.

```
kcc@linux:~

# pwd
/usr/lib/gcc-lib/i386-redhat-linux/2.96
# ls -F
SYSCALLS.c.X   cpp0*                        jc1*
libgcj.spec     libobjc.so.1.0.0@
cc1*             crtbegin.o    jvgenmain*  libgcjgc.a
libstdc++.a@
cc1chill*        crtbeginS.o   libchill.a
libgcjgc.so@     libstdc++.so@
cc1obj*          crtend.o                    libg2c.a
libobjc.a        specs
cc1plus*         crtendS.o     libgcc.a
libobjc.la       tradcpp0*
chillrt0.o       f771*                       libgcj.a
libobjc.so@
collect2*        include/      libgcj.so      @
libobjc.so.1@
```

우의 파일들가운데서 cc1이 진짜 C번역기이다. gcc는 C, C++, Objec C 인가를 검사하고 번역, 련결작업을 진행하며 2진형식의 실행프로그람이 만들어지기까지의 모든 과정을 관리해주는 조정자의 역할만 할뿐이다.

C번역기는 cc1, C++번역기는 cc1plus, Obiect C번역기는 cc1obj, FORTRAN77번역기는 f771이다.

그러면 C언어로 작성한 원천프로그람을 번역하는 방법을 간단히 보기로 하자.

```
kcc@linux:~

# vi hello.c
#include <tdio.h>
int main ()
{
printf ("Hello.\n");
return 0;
}
```

hello.c원천파일을 번역해보자. -o설정값은 출력파일을 지정하는 설정값으로서 hello.c원천파일로부터 hello의 실행파일을 만든다는 의미이다.

```
# gcc -o hello hello.c
```

번역한 파일을 실행하기 위해서는 ./hello를 입력한다. 이것은 현재등록부의 파일을 실행한다는 의미이다.

```
kcc@linux:~
# ./hello
Hello.
```

gdb

Linux에서 사용하는 오유수정기(Debugger)이다. 오유수정기란 어떤 프로그람이 수행되는동안 그 내부에서 일어나는 일을 감시할수 있으며 이 프로그람에 문제가 발생하였을 때에 무슨 일이 일어났는가를 확인할수 있게 해주는 프로그람을 말한다. 이러한 오유수정기인 gdb의 기능은 크게 다음의 4가지로 볼수 있다.

- 프로그람을 시작할 때 특별한 조건을 지정할수 있다.
- 지정한 조건에서 프로그람을 정지시킨다.
- 프로그람이 정지되였을 때에 어떤 일들에 대한 검사가 가능하다.
- 프로그람내부를 변경하여 그에 대한 오유를 수정할수 있으며 다음의 진행사항으로 넘어갈수 있다.

보통 gdb는 C, C++, Modula-2, GNU FORTRAN으로 작성된 프로그람들을 오유수정할수 있다. 그의 실행은 쉘명령행에서 gdb명령으로 시작한다. 시작되면 quit명령으로 gdb를 빠져나오기전까지는 말단으로부터 명령을 읽어들인다. 또한 도움말을 help명령으로 볼수 있다.

```
# gdb
```

인수나 설정값없이 gdb를 기동할수 있지만 일반적으로 gdb는 하나 또는 두개의 프로그람이름을 인수로 주어 실행한다.

```
# gdb program
```

실행가능한 프로그람이름과 core파일을 동시에 인수로 주어 기동할수도 있다.

```
# gdb program core
```

실행중인 프로세스를 오유수정하려면 프로세스ID를 두번째 인수로 지정한다.

```
# gdb program 1234
```

make

Linux에서 C로 프로그램작성을 할 때에 점점 프로그램의 크기가 커지게 되고 이후에 원천을 감당할수 없을 정도가 되여 번역이나 관리상 매우 어렵게 되는 경우가 있다. make는 이것을 위해 프로그람의 일부가 수정된 경우에 필요한 부분만 다시 번역, 련결작업을 하여 개발관리와 유지를 효률적이고 편리하게 하기 위해 사용하는 편의프로그람이다.

보통 프로그람의 량이 많아질수록 모듈별로 나누어 관리하게 되는데 이것을 위해 make편의프로그람은 Makefile 또는 makefile파일에 일정한 형식으로 지정하여 입력파일로 받는다.

이 파일의 내용은 관계되는 파일의 이름과 그 파일에 의존하는데 따라, 즉 그 파일이 변경되였을 때에 그에 따라 함께 변경되여야 할 파일들의 목록과 그 작업을 수행하는 명령문이다.

실례를 들어 설명해보도록 하자. 먼저 foo.c원천파일과 bar.c원천파일은 이미 프로그람으로 작성되여있다고 보고 Makefile만을 만들어서 make에 대한 리해를 가지도록 하자.

```
kcc@linux:~

# vi Makefile
foo:      echo.o hello.o
            gcc -o foo echo.o hello.o
echo.o :  echo.c
            gcc -c echo.c
hello.o:  hello.c
            gcc -c hello.c
```

이 파일을 만들었다면 다음은 번역할 차례이다.

```
# make
```

명령을 사용하거나 또는 아래와 같이 입력하여 번역할수 있다. 이 명령은 gcc -o foo echo.c hello.c를 실행한것과 같은 형식으로 진행된다.

```
# make foo
```

```
kcc@linux:~
# make
make: foo'는 이미 갱신되였습니다.
```

다시 한번 make명령을 실행해보자. 그러면 아래와 같은 통보문을 볼수 있는데 이미 make가 실행되였다는것을 알려 준다. make가 이것으로부터 어떤 파일을 다시 만들어내야 하는가를 알아내는것은 파일이 만들어진 시간을 기준으로 한다. 실례를 들어 echo.o가 만들어진 시간이 echo.c보다 빠르다면 우와 같은 통보문을 출력하는것이다.

```
kcc@linux:~
# vi  hello.c
# include <stdio.h>

int hello()
{
printf ("Hello.\n");
}
~
~
"hello.c" 6L, 57C written
```

원천파일을 수정하여 다시 번역해보자.

```
kcc@linux:~
# make
gcc-c hello.c
gcc -o foo echo.o hello.o
```

hello.c파일이 수정된 다음 다시 번역되는것을 볼수 있다.
다음은 echo.c와 hello.c원천파일의 실례이다.

```
kcc@linux:~

# cat echo.c
extern int hello ();
int main ()
{
hello ();
return 0;
}

# cat hello.c
#include <sidio.h>
int  hello()
{
printf("Hello.\n");
}
```

그러면 마지막으로 Makefile에 목적파일을 삭제하는 문장을 추가해보자.

```
kcc@linux:~

# vi Makefile
foo:   echo.o hello.o
         gcc-o foo echo.o hello.o
echo.o: echo.c
         gcc -c echo.c
hello.o:hello.c
         gcc-c hello.c
clean:
         rm -rt foo echo.o hello.o
~
~
"Makefile"Tk
```

우에서 clean문구를 추가하여 목적파일과 실행파일을 clean명령으로 삭제할수 있게 한다.

make clean으로 번역된 목적파일과 실행파일을 삭제할수 있는것이다.

```
kcc@linux:~

# make clean
rm -rf foo echo.o hello.o
```

patch

프로그람이 이전의 판본에서 일부 수정되였을 때 전체 원천파일을 다시 내리적재받아 처음부터 번역할수도 있지만 변경된 부분만을 적재받아 patch명령을 리용하여 이미 있던 원천코드에 변경내용을 적용할수도 있다. 이 방법은 변화된 부분을 신속히 반영할수 있으므로 매우 편리하다.

다음의 실례에서 만일 hello.c원천프로그람에 새로운 프로그람이 추가되였다고 가정하자.

diff명령은 두 파일의 차이를 비교하는 명령으로서 아래와 같이 -c설정값을 사용하여 hello.c에 대한 새로운 판본을 적용할수 있다.

```
kcc@linux:~
# diff -c hello.c helloc.old
*** hello.c    Tue Jul 23 03:49:59 2004
--- hello.c.old  Tue Jul 23 03:48:42  2004
***************
***2,6****

   int hello()
   {
! printf ("Hello !!! this file was patched ....\n");
   }
---2,6----

   int hello()
   {
! printf ("Hello.\n");
   }
 # diff -c hello.c hello.c.old > hello.patch
```

우와 같이 diff -c명령으로 hello.patch파일을 생성하였다면 부분적인 수정을 할수 있는 파일이 만들어진것으로 된다.

이 .patch파일을 웹브싸이트를 통해 배포하는 경우 hello프로그람을 사용하고있던 사용자들은 직접 hello.patch파일을 내리적재받아 다음의 명령으로 변경적용작업을 실행할수 있다.

```
kcc@linux:~
# patch < hello.patch
can't  find file to patch at input line 1
Perhaps you should have used the -p or —strip option?
File to patch: hello.c
patching file hello.c
Reversed(or previously applied) patch detected! Assume -R? [n] y
```

이와 같이 하여 새로운 원천프로그람을 적용하게 되였다.

python

Python은 Perl과 비슷하게 해석기(Interpreter)를 리용하는 오브젝트지향프로그람작성언어로서 강력한 기능과 함께 문법이 명확하고 리해하기 쉬운것으로 하여 널리 리용되고있다. 이것은 비교적 쉽게 프로그람을 작성할수 있고 C, C++를 사용하여 새로운 모듈을 쉽게 생성할수 있으며 클라스나 례외처리, 고차원동적자료형, 동적인 자료형결정기능 등을 가지고있다. 또한 UNIX나 Linux환경은 물론이고 다른 조작체계들에 대한 이식성이 좋다.

```
kcc@linux:~
# python
Python 1.5.2(#1, Oct 13 2001, 09:06:03) [GCC 2.96 20000731
(RedHat Linux 7.1 2 on linux-i386
Copyright 1991-1995 Stichting Mathematisch Centrum, Amsterdam
>>>
```

Python을 실행하기 위해서는 쉘입력독촉상태에서 python명령을 실행한다. 끝내려면 [Ctrl+D] 건을 입력한다.

만일 Python이 설치되지 않았다면 http://www.python.org에서 제공받을수도 있다.

java

Java(자바)는 인터네트의 분산환경에서 사용되도록 설계된 프로그람작성언어로써 C++언어와 비슷하지만 보다 사용이 간단하고 프로그람작성의 완전한 오브젝트지향성을 강화한 고급해석언어이다. 현재 Java는 많은 개선이 이루어져 JSP, Servlet, EJB, Mobile Java 등과 같은 많은 분야에서 쓰이고있으며 Java개발도구일식인 JDK(Java Development kit)의 Linux용은 무료로 제공되고있다. 일반 배포판들에도 대부분 포함되여있으며 웨브싸이트를 통해 내리적재받아 직접 원천상태로 설치할수도 있다.

이것을 설치하면 Java번역기인 javac, 해석기인 java, 애플리트를 실행하여 보여줄수 있는 appletviewer 등이 추가로 설치되게 된다.

Tcl

Tcl은 UC버클리대학의 존 오우스터호이트가 개발한 해석기형의 스크립트 언어로서 개발 및 유지보수의 책임은 Scriptics회사가 지고있다.

Tcl은 JavaScript, Visual Basic, Perl, REXX 등의 언어들과 비교될 수 있다.

일반적으로 스크립트언어는 보다 구조적이며 번역을 필요로 하는 C나 C++ 언어에 비해 프로그람코드작성을 쉽고 빠르게 할수 있다. 스크립트언어는 때때로 번역된 일부 프로그람들을 서로 묶는 일종의 접착제와 같은 역할을 하기도 한다. Tcl로 GUI를 구현하는데 도움을 주는 프로그람인 Tk라는것도 있다.

Tk

Tk(Tool kit)는 Tcl의 확장판으로서 GUI(Graphic User Interface)를 위한 프로그람작성환경이다. 이것은 확장과 이식이 쉽고 개발시간을 적게 들일수 있게 하며 사용하기가 쉽다. 또한 프로그람작성자에게 표준단추, 안내문, 목록칸(List Box), 화면흐름띠(Scroll Bar), Tcl프로그람작성자가 프로그람개발을 빠르게 하는데 사용할수 있는 두개의 특수한 기능인 Text Wizard와 Canvas Wizard 등의 도구일식을 제공하여 X Window System상에서 대면부구성을 위한 GUI를 간단히 실현하기 위한 제품이다.

Qt

Qt는 C++환경의 GUI서고로서 Windows, Unix, Linux를 지원하며 원천코드의 호환을 보장한다. 이것은 Windows환경에서 만들어진 원천프로그람을 Linux나 Unix환경에 이식하는 시간을 절약해줄뿐만아니라 꼭같은 원천프로그람으로 Windows와 X-Windows실행프로그람을 생산해낼수 있는 우점을 가진다. 또한 Java와 같은 해석기방식의 코드가 아닌 순수한 실행코드를 생성할수도 있다. 이것은 QPL(Qt Public License)하에 사용하는 KDE탁상환경의 X Windows개발도구일식이다.

GTK +

GTK는 원래 GIMP(General Image Manipulation Program)라는 image프로그람을 만들기 위한 도구일식으로 개발되였으며 xlib함수들에 대한 기본적인 Wrapper인 GDK(GIMP Drawing Kit)를 토대로 하고있다.

이것은 GPL사용권한에 따르며 이식성, 오브젝트지향프로그람작성을 지원하는 GNOME탁상환경의 X Window도구일식이다.

Lex & Yacc

lex와 yacc는 벨연구소(Bell Laboratories)에서 1970년대에 개발되였는데 후에 이것들은 UNIX의 표준편의프로그람으로 선정되였다. System V와 초기의 BSD판본에서는 원래의 AT&T판이 리용되고있으며 새로운 BSD판본으로는 flex나 Berkely yacc가 사용되고있다.

또한 FSF(Free Software Foundation)의 GNU프로젝트로 개발된 bison은 yacc에 비해 보다 개선된 기능을 가지고있으며 BSD와 GNU프로젝트의 flex(Fast Lexical Analyzer Generator)외에도 여러 조작체계하에서 다른 이름으로 만들어진것들이 있다.

lex와 yacc는 입력구문의 검색과 분석을 쉽게 하기 위해 개발되였다. 입력에 대한 어떤 패턴을 가져야 하는 응용프로그람을 만들 때에 보다 빠르고 정확하게, 약간의 수정으로 유지보수를 쉽게 할수 있는것이 lex와 yacc의 특징이다.

lex와 ycc는 수자와 연산자의 구별, 번역기나 해석기에서의 문장검색이나 문장해석, 자료기지에서 SQL언어의 문법검사, lex만을 리용한 구분 및 분석등의 여러가지 경우에 리용된다.

색　인

가상뿌리등록부(chroot) 87
가장 널리 사용되는 Apache 웨브봉사기(httpd) 617
같은 내용을 두 경로로 동시에 출력(tee) 526
건반설정(kbd_mode) 260
건반입력속도의 설정(kbdrate) 261
건반코드값보기(dumpkeys) 145
검사(test) 531
경로이름보기(dirname) 133
경로표보기(route) 453
공백을 tab로 변환(unexpand) 557
구동프로그람설정(chkconfig) 74
구획설정(cfdisk) 63
구획설정(fdisk) 165
구획의 남은 용량을 보기(df) 127
집단변경(chgrp) 72
집단변경(newgrp) 380
집단삭제(groupdel) 204
집단정보수정(groupmod) 205
집단추가(groupadd) 203
그림자집단생성(grpconv) 208
그림자암호의 사용(pwconv) 422
기동리력의 보기(dmesg) 134
기본쉘의 호출(sh) 479
기억등록부의 보기(dirs) 663
개행문자변환(col) 93
게트웨이경로지정(gated) 195
관련명령의 찾기(apropos) 9
권한변경(chmod) 77

날자보기(date) 116
내부명령문도움말(help) 668
내용정렬(sort) 489
다른 식별이름의 환경을 사용하기(su) 509
다른 사용자에게 통보문 보내기(write) 579
다중분기문(case) 661
단어수의 계산(wc) 570
동적련결의 설정(ldconfig) 275
동적 IP 주소의 제공(dhcpd) 615
등록부만들기(mkdir) 346
등록부삭제(rmdir) 449
식별이름 ID 의 확인(id) 227
디스크사용용량의 제한(quota) 424
디스크사용정형을 보기(du) 141
대기(sleep) 486
력서보기(cal) 50
련결원본파일의 경로보기(namei) 375
리력파일관리(logrotate) 300
마운트정보보기(showmount) 482
말단을 초기화하는 명령문(reset) 439
말단이름보기(tty) 546
말단의 작업기록(script) 474
말단환경설정(stty) 504
망구동스크립트(network) 629
망대면부설정(ifconfig) 229
망련결확인(arping) 16
망련결확인(ping) 402
망상태보기(netstat) 377
명령검색(whatis) 572
명령문경로의 검색(whereis) 573
명령문을 간단히 사용하기(alias) 2

명령실행의 예약(batch) 42
모듈의존성의 확인(depmod) 126
모듈제거(rmmod) 450
모든 사용자에게 통보문 보내기(wall) 569
목적파일의 크기축소(strip) 502
문서보기(man) 327
문서변환(expand) 158
문자렬검색(grep) 201
문자렬찾기(strings) 500
문자의 변환 및 삭제(tr) 542
문자확대출력(banner) 38
밑줄긋기(ul) 549
밑줄(__)문자변환(colcrt) 94
매 시간대에 의한 시간보기(zdump) 599
변수지정(export) 665
본문계산기(bs) 44
불량블로크의 검사(badblocks) 35
블로크단위의 복사(dd) 119
빈 파일을 만들기(touch) 540
사본만들기(cpio) 105
사용자보기(users) 567
사용자보기(who) 576
사용자정보보기(finger) 178
사용자정보의 변경(chfn) 70
사용자정보의 변경(ypchfn) 587
사용자의 추가(newusers) 382
사용자의 추가(useradd) 561
사용자의 제거(userdel) 563
사용자설정환경의 변경(usermod) 564
사용중인 파일들의 목록보기(lsof) 319
산수연산(let) 673

상태보기(stat) 497
서고변환(ranlib) 429
서고의존성을 보기(ldd) 277
소유자변경(chown) 84
속성변경(chattr) 68
시간맞추기(rdate) 432
신호기파일생성기(lockfile) 292
실행준위의 보기(runlevel) 471
실행파일압축(gzexe) 211
쉘변경(chsh) 88
쉘선언(#!) 658
쉘압축편의프로그람(shar) 480
쉘의 변경(ypchsh) 588
암호유효기일의 지정(chage) 65
암호의 변경(passwd) 395
암호의 변경(yppasswd) 591
암호파일검사(pwck) 421
압축된 본문파일의 내용보기(zmore) 600
압축된 파일의 내용을 보기(zcat) 596
압축된 파일의 내용비교(zcmp, zdiff) 598
압축명령문(bzip2) 47
압축해제(gunzip) 209
압축해제(uncompress) 556
우선권순위변경(nice) 383
우선권순위변경(renice) 438
우편을 모아오기(fetchmail) 171
우편도착의 알림(biff) 46
우편봉사기데몬(sendmail) 636
우편의 송수신(mail) 321
우편작업보기(mailq) 326
우편 alias 보기(praliases) 408

유효사용자의 보기(whoami)..578
이름봉사기데몬(named) 622
이름봉사기문의(nslookup)..389
이전 명령문(history) 669
인쇄기관리자(lpc) 303
인쇄대기렬의 보기(lpq) 305
인쇄명령(lpr) 307
인쇄취소(lprm)..309
인터네트고급데몬(xinetd) 647
읽기전용(readonly) 436
예약작업보기(atq) 23
예약작업삭제(atrm) 24
예약작업의 실행(atd) 611
예약작업의 실행(crond)..613
완료(exit)..664
원격실행(rexec)..442
원격호스트 쉘을 사용(rsh) 468
원격호스트접속(rlogin) 444
원격우편보기(rmail) 448
원격접속(telnet) 528
원격접속(ztelnet) 603
원격지파일복사(rcp) 430
자기의 집단보기(groups) 206
작업예약(at)..18
장치련결(mount)..366
장치련결의 해제(umount)..550
장치조사(rdev)..433
적재된 모듈의 보기(lsmod) 318
정지(break)..659
접속리력의 보기(lastlog) 272
접속자확인(rwho) 473

주해(#) 657
직렬회선련결(slattach) 485
주기억기의 내용을 보관(sync) 514
중복행의 추출(uniq) 558
지역정보보기(locale) 289
참(0)값을 반환(true) 545
철자법검사(ispell) 256
초기화프로쎄스조종(init) 236
체계식별이름정보의 제공(rstat) 470
체계리력의 기록(logger) 294
체계설정(setup) 478
체계이름보기(hostname) 225
체계에 발생하는 다양한 리력들을 기록하는 봉사(syslog) 644
체계의 MAC 주소확인(arp) 13
체계완료(halt) 214
체계정보의 보기(uname) 552
체계주기억의 보기(free) 180
체계접속명령문(login) 296
체계접속시간의 보기(last) 269
체계재기동(reboot) 437
체계탈퇴중에도 프로그람실행(nohup) 388
체계끌내기(shutdown) 483
코드대응표보기(getkeycodes) 200
통보문보내기(rwall) 472
특수파일만들기(mknod) 352
특정렬의 삭제(colrm) 96
특정문자렬로 시작하는 행을 보기(look) 302
파일경로의 검색(which) 575
파일내용을 거꾸로 바꾸기(rev) 441
파일내용을 역순으로 출력(tac) 515
파일등록부이동(cd) 62

파일련결명령문(ln) 286
파일류형확인(file) 173
파일목록보기(ls).. 311
파일묶기(tar).. 522
파일보기(cat).. 58
파일보기(less).. 278
파일복사(cp) 103
파일복사(scp) 476
파일분할(csplit).. 111
파일분할(split) 493
파일비교(cmp) 92
파일삭제(rm).. 446
파일설치(install).. 242
파일속성보기(lsattr) 316
파일압축(compress) 101
파일압축(gzip) 212
파일앞부분의 보기(head).. 219
파일이동(mv) 374
파일이름검사(pathchk) 399
파일이름추출(basename).. 40
파일의 다른 점을 비교(diff).. 129
파일의 뒤부분보기(tail) 517
파일인쇄설정(pr) 405
파일위치찾기(locate) 291
파일정렬(column).. 98
파일찾기(find) 175
파일찾기(grpck).. 207
파일찾기(slocate) 487
파일체계검사(fsck) 182
파일체계설정(tune2fs).. 547
파일체계정보의 보기(dumpe2fs).. 143

파일체계조종(debugfs) 122
파일합치기(paste) 397
파일확인(sum) 511
파케트이동경로 보기(traceroute) 544
평균부하의 보기(uptime) 560
표준입력을 명령으로 변환(xargs) 580
표현식평가(expr) 160
프로그람에서 실행되는 프로세스의 추적(strace) 499
프로세스상관도(pstree) 419
프로세스상태보기(ps) 410
프로세스상태 보여주기(top) 536
프로세스완료(kill) 262
프로세스완료(killall) 265
프로세스식별(fuser) 194
프로세스조종(telinit) 527
플로피디스크의 초기화(fdformat) 163
플로피디스크설정(setfdprm) 477
패턴처리언어(awk) 26
하드디스크정보의 보기 및 설정(hdparm) 216
하드웨어정보의 보기(lsdev) 317
하이퍼본문형의 문서페지(info) 234
학습장형콤퓨터의 전원관리(apm) 5
한행씩 비교(comm) 99
항목단위로 파일합치기(join) 258
항목의 골라보기(cut) 114
항상 실패(false) 161
현재경로보기(pwd) 423
현재 사용자이름의 보기(logname) 298
현재 사용자의 보기(w) 568
호스트정보보기(host) 222
호스트 ID지정(hostid) 224

핵심부모듈을 불러들이기(insmod) 239
핵심부모듈적재(modprobe).......................... 360
핵심부 symbol의 보기(ksyms) 268
행번호매기기(nl) 386
행인쇄기데몬(lpd) 619
행편집기(ex) 157
화면단위로 파일보기(more).......................... 364
화면지우기(clear) 91
환경변수출력(echo).......................... 150
환경변수보기(env).......................... 155
환경변수보기(printenv) 409
환경설정의 반영(source).......................... 492
뿌리장치설정명령문(rootflags) 452
awk-perl 변환(a2p) 33
CD-ROM 자동반출(eject) 153
CPU 형의 보기(arch).......................... 11
crontab 관리(crontab) 108
DNS 령역확인(dnsdomainname) 137
dump 예비보관파일을 복구하는 명령문(restore) 440
ext2 파일체계를 생성하는 명령문(mke2fs).......................... 347
ftp 데몬(proftpd) 632
ftp 봉사기접속의뢰기프로그람(ftp) 187
ftp 사용자확인(ftpcount) 192
ftp 접속자보기(ftpwho).......................... 193
for 순환(for).......................... 666
gated 관리(gdc).......................... 197
gcc 703
gdb.......................... 707
GTK +.......................... 718
if 조건문(if) 671
java 714

Java 애플리트의 실행(appletviwer) 7
Kernel 2.2.x 판본의 방화벽관리도구(ipchains) 243
Kernel 2.0.x 판본의 방화벽관리도구(ipfwadm) 249
Kernel 2.4.x 판본의 방화벽관리도구(iptables) 251
key 값의 보기(ypmatch) 589
Lex & Yacc 719
Linux 기동적재기(lilo) 281
Linux 파일체계검사(e2fsck) 147
Linux 파일체계생성명령문(mkfs) 350
Linux-DOS 간의 파일복사(mcopy) 335
make 708
MS-DOS 등록부변경(mcd) 333
MS-DOS 등록부삭제(mrd) 370
MS-DOS 디스크식별표식만들기(mlabel) 355
MS-DOS 파일내용보기(mtype) 373
MS-DOS 파일목록보기(mdir) 341
MS-DOS 파일삭제(mdel) 337
MS-DOS 파일삭제(mdeltree) 339
MS-DOS 파일이동/이름변경(mmove) 359
MS-DOS 파일이름변경(mren) 371
MS-DOS 파일의 속성변경(mattrib) 330
MS-DOS 파일체계등록부의 만들기(mmd) 357
MS-DOS 파일체계만들기(mformat) 343
MS-DOS 파일체계의 복구(dosfsck) 140
MS-DOS 파일체계의 불량블로크 표시(mbadblocks) 332
MS-DOS 파일체계에서 매개변수보기(minfo) 345
MS-DOS 파일체계적재(mmount) 358
NFS 봉사기데몬(nfs) 631
NIS 구동(ypbind) 582
NIS 련결의 확인(ypcat) 585
NIS 령역의 보기(domainname) 138

NIS 봉사기(ypserv) 594
NIS 전체의 반영(yppush) 593
NIS 정보의 보기(yppoll) 592
OpenSSH 봉사기데몬(sshd).. 642
patch.. 711
PCMCIA 기판의 관리(cardctl) 53
PCMCIA 장치관리자(cardmgr)... 56
PID 확인(pidof) 401
python... 713
Qt 717
RAM 디스크크기의 확인 및 설정(ramsize) 427
RAM 변환명령(rpm2cpio) 467
RedHat 패키쥐관리자(rpm) 456
Samba 봉사데몬(smb).. 640
SSH 의뢰기(ssh) 496
swap 령역지정(mkswap)... 354
swap 설정(swapon) 513
swap 중지(swapoff).. 512
Tcl 715
tftp 봉사기접속의뢰기프로그람(tftp)................................ 534
Tk.. 716
while 순환문(while) 675
1 대 1 의 대화(talk) 520
8 진수로 파일보기(od) 393
.gz 로 변환(znew) 602

UNIX, LINUX 명령어사전

편 역: 김철호, 김광철, 김용규, 리일화, 오주혁, 안성범, 김 훈
편 집: 안성범, 오주혁, 김은영, 최은심, 리일화, 김희란, 김 훈
편 성: 허경심, 리영순, 김은심, 량혜련, 배혜경, 박혜련, 류 철
정성진, 최태영
장 정: 전형란, 오국현
심 사: 정성봉, 박주학, 리일화, 김희란, 오주혁, 김 훈

낸 곳: **정보산업지도국**

제작기관: **정보기술출판소**

발 행: 주체 94(2005)년 10 월 10 일